Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Autoencoder Based Anomaly Detection of Electricity Theft in House Hold Consumer side of the smart grid

Jadhav Girish Vasantrao [1], Dr. Chirag Patel [2]
[1]*Parul University,* [2]*Charotar University of Science and Technology*
Email Id: [1]*girish.jadhav@paruluniversity.ac.in,* [2]*Chiragpatel.dce@charusat.ac.in*

ARTICLE INFO

Received: 29 Dec 2024

Revised: 12 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

Introduction: Theft of electricity continues to be an ongoing problem with serious implications, such as loss of revenue, grid instability, decreased efficiency, and higher likelihoods of system overloads. The covert operation of this act presents a tremendous challenge to global power distribution networks, both to utility companies and consumers as energy needs and expenses keep on growing.

Objectives: The objective of this study is to establish a consistent method for identifying electricity theft in a 19-bus power distribution system. The research targets the detection of energy usage anomalies that could be a sign of fraud.

Methods: The suggested method involves the use of Long Short-Term Memory (LSTM) Autoencoders, which have proved to be efficient in detecting anomalies. The model combines LSTM and Autoencoder techniques for handling time-series data. The method involves creating input sequences, an LSTM Encoder and Decoder, and using anomaly detection methods.

Results: Through model training and anomaly detection, the research renders essential energy theft patterns with the help of simulations. The performance analysis demonstrates the strength of the suggested model for anomaly detection in the power distribution system.

Conclusions: The results indicate that LSTM Autoencoders present a sound framework for electricity theft detection. The proposed method may be applied in different real scenarios, and they can help towards building more reliable and efficient power distribution systems.

Keywords: Anomaly detection, LSTM Autoencoders, power systems, and simulations.

INTRODUCTION

The issue of electricity theft poses a significant challenge in power systems due to its substantial financial implications. Electricity thefts have a detrimental influence on the functioning of the power grid with respect of both its durability and its efficacy, in addition to the fact that they cause the system to become overloaded. Theft of electricity, a clandestine and unlawful activity, poses a significant challenge to power distribution systems worldwide. With the increasing demand for electricity and the rising costs of energy, electricity theft remains a persistent issue that impacts both utilities and consumers. In the context of complex power distribution systems, such as a 19-bus network, detecting electricity theft is not only a matter of economic concern but also a critical aspect of maintaining the integrity of the grid[1]. The authentication software of the smart metres makes it possible for malevolent users to undermine the integrity of the power consumption data in order to reduce their monthly electricity bills. Theft of energy has a multiplicity of negative repercussions: it results in revenue losses for utility companies, it overloads the distribution network, and it creates potential safety problems[2]. It also threatens the reliability of the power supply for legitimate users and, more generally, it stops the environmentally friendly growth of electrical infrastructure.[3].

In order to address this widespread problem, it is necessary to employ innovative and precise anomaly detection techniques. Conventional approaches to detecting instances of electricity theft frequently prove insufficient, particularly in the context of wide power distribution systems. These systems show complicated consumption patterns and grid characteristics that require the implementation of advanced solutions. The research endeavors are centered around addressing the challenge through the utilization of LSTM Autoencoders, an advanced neural

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

network architecture. The main goal is to use this technology for the identification of power theft[4]. To counteract this widespread problem, novel and accurate anomaly detection approaches are necessary. Traditional approaches for detecting energy theft often fall short, particularly in large-scale power distribution systems with complicated consumption patterns and grid features that need specialised solutions[5].

In response to this problem, the paper research focuses on the development and use of LSTM Autoencoders, a sophisticated Neural Network architecture, for the goal of detecting power theft. To overcome this issue, LSTM Autoencoders provide a strong and adaptable tool[6]. Because of their capacity to incorporate temporal connections within sequential data, these neural networks are well-suited for analysing power use trends. The LSTM architecture is a (RNN) expansion meant to aid with both short-term and long-term memory retention[7]. Unlike standard RNNs, which may only remember knowledge up to a given point in time, LSTM networks may maintain a whole history of prior information. [8]

The study aims to introduce a robust method for detecting energy theft in a 19-bus bar power distribution system using LSTM Autoencoders, focusing on identifying anomalies in energy usage trends indicative of potential theft or abuse.

The paper's succeeding parts are organized as: Section II explains the LSTM Autoencoder architecture and its use for anomaly detection. Section III discusses the associated work. The approach used is described in Section IV, which includes data preparation, model training, and anomaly detection. Section V displays the acquired findings, which demonstrate the performance and efficacy of the suggested strategy. Finally, Section VI summarizes the results and emphasizes the LSTM Autoencoder-based anomaly detection importance in tackling energy theft concerns.

BACK GROUND

A. LSTM

The LSTM stands for "long short-term memory," and it is often seen as a development of "Recurrent Neural Networks" (RNN). It is able to offer a "short-term memory" capacity, which enabled the retrieval of previously stored knowledge (but only up to a specific point) for the purpose of performing the current job. LSTM architecture is an extension of RNN and gives the ability often referred to as "long-term memory." This suggests that the present neural node may access a list of all the data that has been collected in the past, as opposed to simply one specific point in time. The LSTM unit shown in Figure 1 is consisting of the cell, the forget gate, the input gate, and the output gate respectively. Long-term memories are stored in cells, and the activity inside these cells is governed by three gates that regulate the information that enters and exits the cell out of the cells:

• The concept of "Cell State" describes the active long-term memory of the network at the current time, which preserves a historical record of information that has been received in the past.

• The output that is denoted by the "previous Hidden State" is the one that was produced by the "previous time step," which operates as a form in terms of short-term memory.

• The value of the input measured at the most recent time step is included in the input data.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

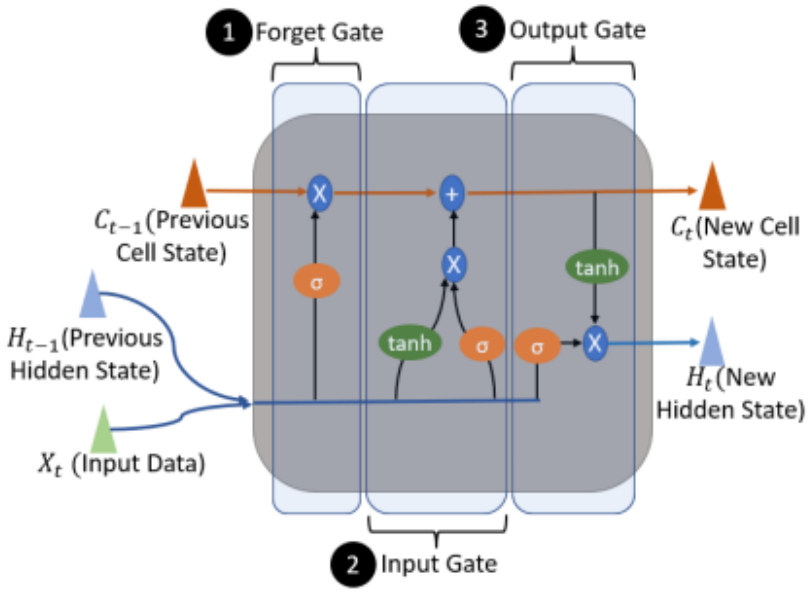

Figure 1 working of LSTM

Step1 : Forget Gate

The forget gate is mainly intended to serve the function of ascertain the relevance specific's bits within the cell state in light of both the information that was previously buried and the new data that has been supplied. By employing a sigmoid activation function, to do this, the previously concealed status as well as the newly acquired input data are both taken into account by the neural network. The Neural Network (NN) constructs a vector with each element normalized to the interval [0,1]."

When a section of the input is determined to be unimportant, the forget gate component of the network is taught to provide outputs that are near to zero; otherwise, it produces outputs that are closer to one. Following that, the state of the cell that came before them is multiplied point-wise by these outcomes as they are communicated up. The findings of f_t) originating from the forgotten gate might be given in the following mathematical format:

$$f_t = \sigma\left(\omega_f \left[H_{t-1}, X_t\right] + b_f\right) \qquad (1)$$

In this σ represents the function of activation the, ω_f and b_f stand for forget gate's weight and bias, respectively. The concatenation of hidden state is represented by H_{t-1}, while the concatenation of the current input is represented by X*t*.

Step2 : Input Gate

The input gate serves two primary functions in its overall design. The first step is to determine whether or not it is beneficial to maintain the new information (i.e., the previously concealed state as well as the new input data) in the cell state. Secondly, the system determines the appropriate additional information to be incorporated into the cell state, depending on the presence or absence of certain conditions. In order to do this, the input gate will go through two separate procedures. Combining the prior hidden state with the newly entered data is one step in the process, which results in the generation of a new memory update vector, which is denoted as C_t. The memory update vector's components are created using the *tanh* activation function, which makes sure that they are between [-1, 1]. To lessen a component's influence on the cell state, negative values are used. The aforementioned elements possess value due to their utilisation of the *tanh* activation function for their generation. Given the new information, this vector indicates the degree to which each component of the cell state should be updated. Equation 2 provides a representation of this process, which may be found below.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$$C_t = \tanh\left(\omega_c \left[H_{t-1}, X_t\right] + b_c\right) \qquad (2)$$

In which ω_c weight matrices and b_c stand for the bias of the input gate, both of which are respective, and the *tanh* is activation function.

The determination of whether the elements of the incoming input are worth remembering is an important aspect of the input gate phase, especially within the context of the veiled condition. Incorporating the use of the sigmoid activation function, a vector of values in the interval [0,1] is what the input gate is taught to output. This is done in a manner that is similar to the forget gate. The cell state will not be changed with any output that is closer to 0 than it now is. Equation 3 is used in the following to illustrate the procedure.

$$i_t = \sigma\left(\omega_i \left[H_{t-1}, X_t\right] + b_i\right) \qquad (3)$$

a situation in which weight matrices as well as input gate biases are denoted by ω_i and b_i respectively.

These two processes are expanded together point-by-point. This results in the size of the newly acquired information, as determined by Equation 3, being controlled and, if necessary, having its value set to zero. Following this, the combined vector that was produced is added to the state of the cell, which ultimately results in an update to the LTM of the network, as presented by Eq-4.

$$C_t = f_t \square\ C_t - 1 + i_t \square\ C_t . \qquad (4)$$

Step3: Output Gate

It's time to deal with output gate now that the LTM has been updated. The determination of a new concealed state is the primary responsibility of the output gate. In order to do, the output gates make utilize of 3 distinct pieces of details: the most recently updated state of the cell, the most recently concealed state, and the most recent information that you have entered. The now signal is subjected, first, to the sigmoid-activated filter available input data, followed by the previously concealed state network as illustrated in Eq-5, in order to produce the filter vector O_t.

$$O_t = \sigma\left(\omega_0 \left[H_{t-1}, X_t\right] + b_0\right) \qquad (5)$$

Where ω_o and b_o are the weight matrix and the bias of output gate.

After a tanh activation function is responsible for the processing of cell status data are compressed within the range [-1, 1] to produce a squashed cell state. This cell state is then pointwise multiplied with the filter vector to get the final result. As can be seen in Equation 6, a new hidden state, denoted by the letter H_t, is generated and resulted with new cell state, denoted by the letter C_t.

$$H_t = o_t \square\ \tanh(C_t) \qquad (6)$$

In The present LSTM unit, also known as the current cell state, which is indicated by Ct, is transformed into the former cell state, which is marked by Ct-1. The next LSTM unit will perform a conversion that will change the current hidden state, which will be designated by Ht, into the prior hidden state, which will be represented by Ht-1. This won't be complete until the LSTM cells of the system have processed each and every piece of input data arriving from the time series sequences procedure is repeated.

B. Auto encoder

To efficiently learn new information, one may make use of a kind of unsupervised neural network called an autoencoder codings from unlabeled data input. These codings may then be utilised in other applications. This is accomplished by training the neural network to effectively eliminate additional data, commonly known as 'noise'. This allows it to acquire the ability to learn a certain representation for a specific collection of input data. In addition

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

to the input layer and the output layer, a typical autoencoder will also include a great deal of information in the form of hidden layers, and the input layer in addition to those two levels. As shown in Figure 2, the processes that an autoencoder goes through may be broken down into three categories: encoding, decoding, and reconstruction loss.

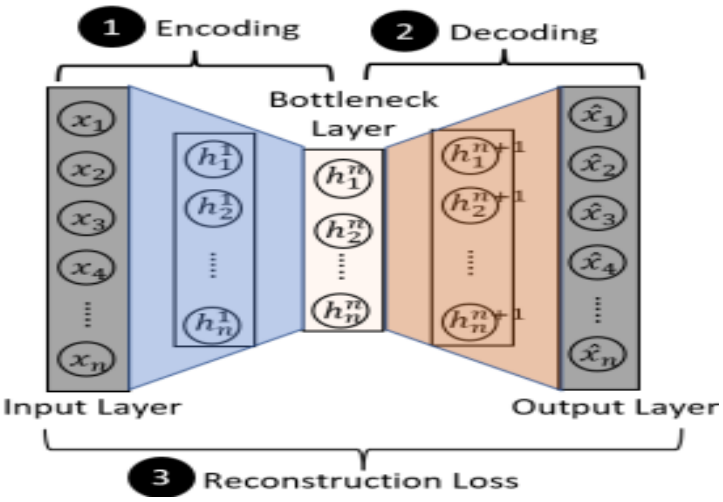

Figure 2 Working of Autoencoder

Step1: Encoding

The encoding procedure begins with a high-dimensional vector representation of the input data (hence x). with m dimensions ($x \in R^m$). This vector is then transferred to a low-dimensional representation of the bottleneck layer, which is indicated by the letter (h), after any unnecessary features have been removed as illustrated in Eq-7.

$$h = f_1\left(\omega_i x + b_i\right) \qquad (7)$$

ω_i stand for the weight matrix, which is sometimes abbreviated as bi, is a shorthand for the and bias; f_1 is the abbreviation for an activation function respectively.

Step2: Decoding

During the decoding process, x is constructed from the representation of (h) at the bottleneck layer. This, in turn, finally results in the reconstruction of x, it is seen by Eq-8

$$x = f_2\left(\omega_j h + b_j\right) \qquad (8)$$

where f_2 refers to the decoder activation function that is currently being used. ω_j denotes the weight matrix, The term "bias" is denoted by the letter bj, while "x" denotes the reconstructed a sample to be entered. Note the following: ω_j and b_j could not be related to the encoder's matching ω_i and b_i values at all.

Step3: Reconstruction Loss

In the context of conventional As shown in Equation 9, A reconstruction loss, denoted by the letter "L," is calculated in order to reduce the gap between the output and the input. The cost of rebuilding is often underestimated used for the purpose of anomaly detection.

$$L(x-x) = \frac{1}{n}\sum_{n=1}^{n}\left|x_t - x_t\right| \qquad (9)$$

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

In this the data entered is denoted by the letter x, the data that is created by the algorithm is denoted by the letter x', and n refers to the number of samples that are included in the dataset that was used for training. On the other hand, this is developed further in such a way that the model is able to determine the reconstruction of a sample that was lost as shown below:

$$x_i = \frac{1}{n}\sum_{n-1}^{n}\left|x_i - x_i\right| \qquad (10)$$

$$s.t. \quad n = \begin{cases} i & \leq \dfrac{N+1}{2} \\ N-i+1 & n > \dfrac{N+1}{2} \end{cases}$$

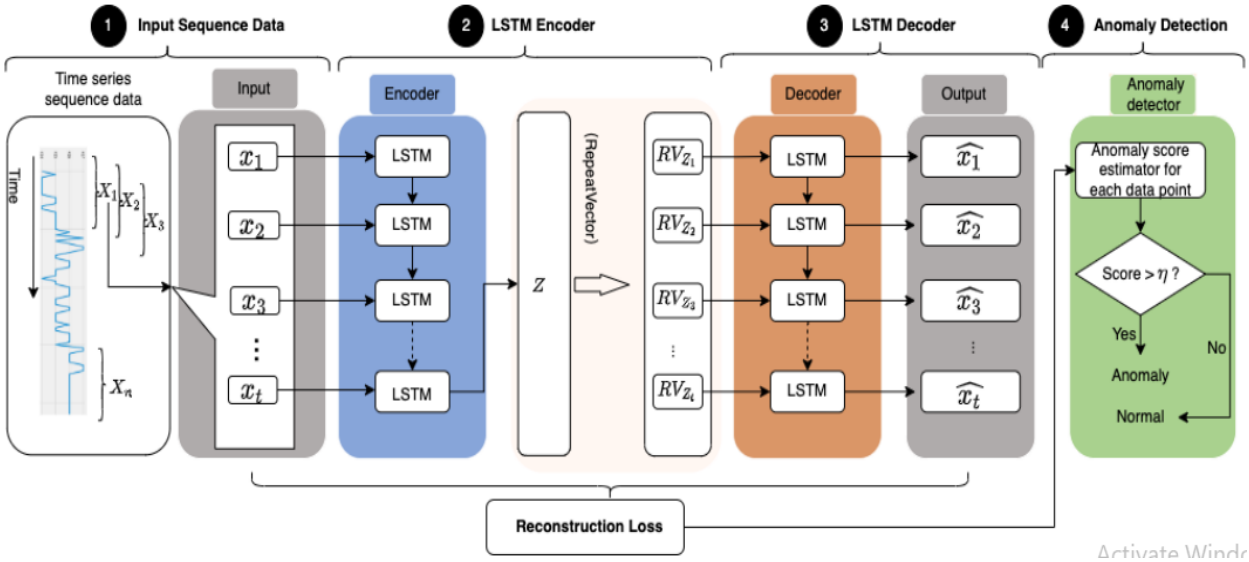

Figure 3 Summary of proposed model

Xi = [x1,..., xi], where N refers to the total number of samples and n indicates the number of samples that are being used. where N refers to the total amount of samples. where n refers to the total number of samples. After that, the process that use to figure out the reconstruction loss for each and every sample that is a part of the time series is as outlined in the following paragraphs.

$$Loss = \frac{1}{N}\sum_{N=i}^{1} x_i \qquad (11)$$

where the overall number of specimens is indicated by the by N and the reconstruction loss that is calculated for each individual sample is denoted by x.

RELATED WORK

In paper [9] discussed smart metres' susceptibility to cyber assaults and presents an anomaly detection system for detecting power theft as well as malfunctioning metres using consumption data. To assess technical losses and measurement noise, it adds metrics such as a factor of loss and as an error word. The framework has been successful in both simulations and test rig operations is proven to successfully discover fraudulent users and detect malfunctioning smart metres. In paper [10], a novel unsupervised machine learning strategy for identifying fraudulent customers utilising the ROBPCA and ORC algorithms, therefore conserving utility resources. It outperforms nine frequently used outlier identification techniques with a high accuracy of 94.34% and a detection

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

rate of 92.52%. In paper [11] introduced an anomaly-based approach to power theft detection using smart metre data, which can be trained with only typical use patterns. When put to the test in the real world, it performs well, has an A mean *F1* value of 0.93 and an average detection lag time of 19 days on average for certain kinds of attacks. The paper in [12] addresses electricity theft detection in the context of cyber threats to smart grids, highlighting the benefits of PCA-based semi-supervised anomaly detection over P2P, with PCA showing a significantly lower false alarm rate (4%) and a 45% improvement in detection accuracy compared to P2P, providing a more robust and effective approach.

In [13] paper offer an Anomaly detection with LSTM Autoencoders system for energy management, with a goal of predicting and preventing energy consumption abnormalities. It detects deviations from regular data patterns while accounting for variables such as weather and time, and it displays its capacity to proactively avert defects and improper operations via experimental validation. Using an unsupervised the study offers an end-to-end autoencoder that is based on long-term and short-term memory anomaly detection method for decreasing energy waste in manufacturing. Industrial applications, unlike residential structures, have not yet been widely addressed for identifying malicious energy use abnormalities in paper [14]. In paper [15] investigates anomaly identification in advanced metering infrastructures utilising deep Autoencoders (BAEs) with LSTM and FC structures, which outperform benchmark approaches such as one-class SVM and ARIMA. The deep LSTM-based BAE design improves true positive rate by 8-9% and reduces false positive rate by 7-16%.

In paper [16] research describes a unique method for detecting early abnormalities in power usage utilising a two-stage LSTM-based neural network and autoencoder system that distinguishes between local and global anomalies. It combines meteorological data, temporal characteristics, and feature selection, resulting in improved detection performance and efficiency, as well as a large rise in consumption during atypical times. The paper proposed a robust anomaly detection system based on deep LSTM on time-series data that has been evaluated using real-world Siemens Industrial Turbomachinery data and provides accurate predictions for a variety of domains. In another paper [17] a method employs autoencoder-based feature selection and encoding, resulting in accurate anomaly identification and prediction. Deep neural networks outperform traditional neural networks, particularly in complicated data settings.

METHODOLOGY

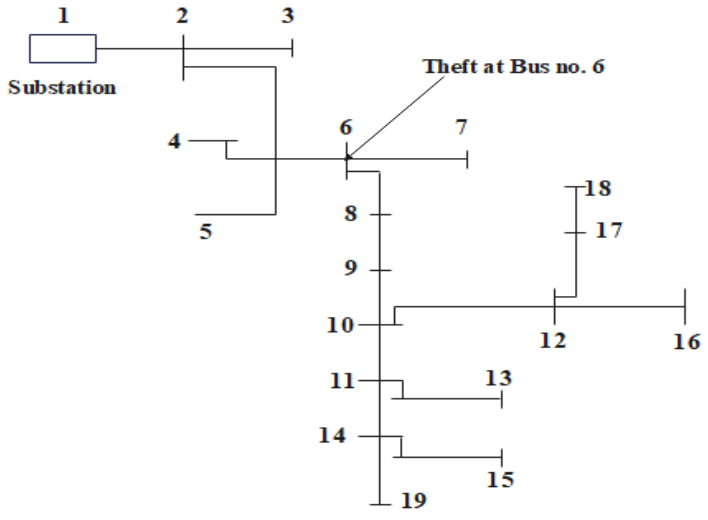

Figure 4 19 bus distribution feeder

The Figure-4 demonstrated the System Architecture of 19 bus feeder distribution system and the theft is detected at bus bas 6 which is indicated in in Figure.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

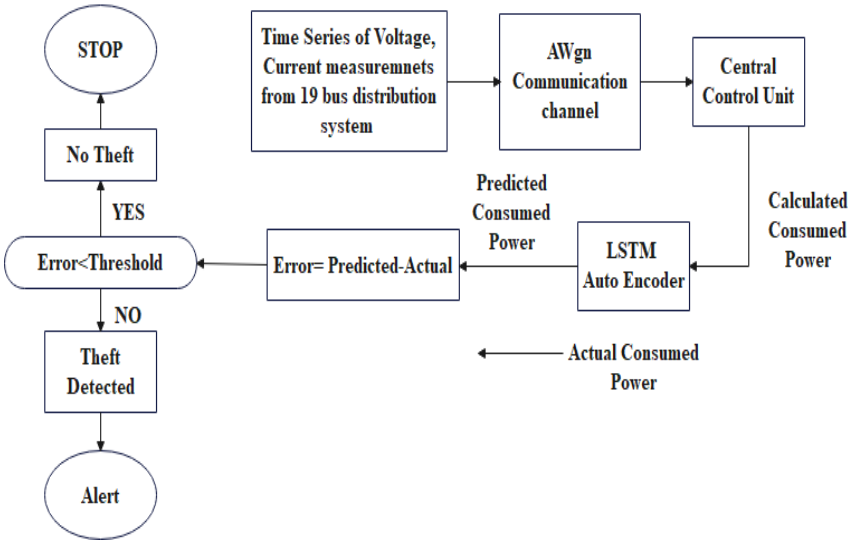

Figure 5l Block Diagram of Proposed Model

The provided flow diagram presents a comprehensive system designed to monitor and detect anomalies in power consumption within a distribution system. The flow diagram illustrates the process of obtaining voltage and current measurements from a 19-bus distribution system over a period of time. These measurements are then transmitted through an Additive White Gaussian Noise (AWGN) communication channel through central control unit. The information is sent to a central processing unit, where it is processed it is utilised to determine energy use. The calculated power is subsequently inputted into an LSTM Auto Encoder, a neural network model that effectively processes and analyses the data. The LSTM Auto Encoder produces a forecast of the power consumption, which is then compared to the observed power consumption. The mistake, which stands for the difference between what was expected and what really happened power consumption, is computed. If the error value is found to be lower than the predetermined threshold, it can be inferred that there is no evidence of theft or abnormality in the power consumption. In the event that the error surpasses the predetermined threshold, it indicates a disparity between the projected and real power usage, suggesting the possibility of unauthorized power consumption or an irregularity. In such instances, an alert is activated, enabling prompt identification and subsequent examination.

In this methodology, authors for suggested model for analysing time-series data, which makes use of a mix of LSTM and Autoencoder to identify abnormalities in the data. First, will provide an summary of suggested model, which is based on the four processes of constructing the input sequence, using the LSTM Encoder and LSTM Decoder, and using the LSTM Encoder and LSTM Decoder, and using the LSTM Encoder and LSTM Decoder, the output sequence is created and LSTM detecting anomalies. In addition, authors provide a comprehensive explanation of the method that the model employs, both in terms of the learning and validation stages.

A. LSTM-Autoencoder

An exemplification of the model under consideration is displayed in Figure 3. The LSTM-Autoencoder builds LSTM networks built on top of the techniques of encoding and decoding used by the Autoencoder, using the capabilities of the Autoencoder in both cases LSTM neural network and the Autoencoder. The encoder will generate a vector with the high-dimensional input data sequence in it, with a predefined size. The LSTM memory cells are used to process the data using the encoder method. This guarantees the preservation of relationships between many points of data in the form of a the progression of events across time. The encoder approach will, in addition to this, continue to convert the high-dimensional input vector representation into a low-dimensional representation up to the point when it enters the latent space. In order to recreate the output, the decoder LSTM employs a condensed version the information from the input data that is kept in the hidden area fixed-size input sequence. A reproduction process threshold is established based on reconstruction error rates. By employing this threshold, one may detect a divergence from the norm.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Step1: Input Sequence Data

The notations [X_1, X_2, X_3,..., Xn] designate a string of time sequences that make up the first dataset. Every sequence X is produced with a data set for a time frame that looks like this: [x1, x2, x3,..., xt], where each m-features that are entered during the time-instance *t* are reflected here by an $X_t \in R^m$ value. *T* is used to represent the length of the time frame. After that, it is reorganized into a two-dimensional array, this time representing samples and timesteps rather than the original timesteps and samples. For instance, a string containing the measured power values and turn it into a two-dimensional array, with each dimension containing a list of the samples that are gathered at ten different times.

Step2: LSTM Encoder

The LSTM encoder's primary function is to simulate a sequence folding layer by converting attributes into a group of feature sequences that are dependent on time. The LSTM encoder's primary objective is to achieve this. It is comparable to performing convolutional operations on individual timesteps of feature sequences in a standalone fashion. Figure 5 provides a detailed explanation of the interactions that take place between the AE encoder as well as the succession of LSTM unit cells that have been taught to detect the most essential components of the input sequence.

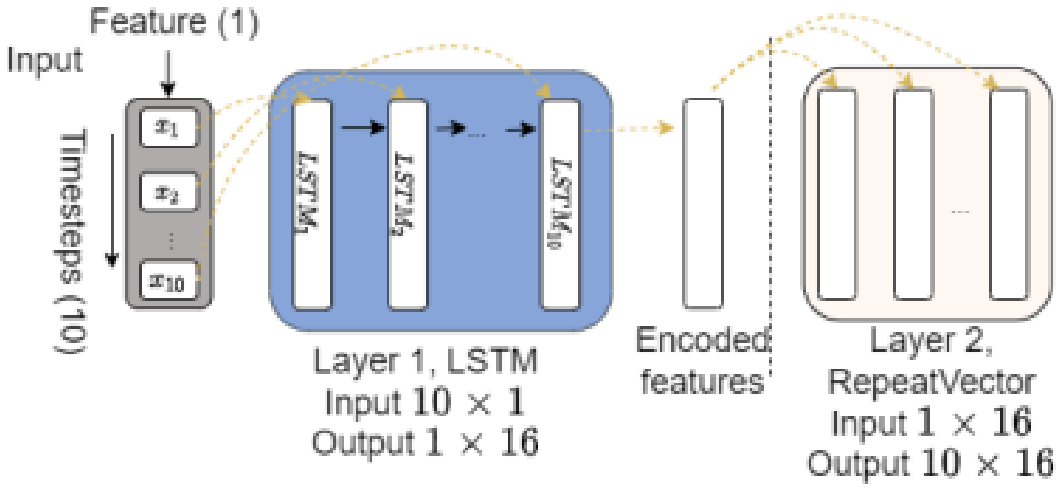

Figure 6 Details of LSTM Encoder

Ten samples, each with a one-minute interval, are gathered over the course of ten timesteps to create each time series of variable X_i. To be entered into the encoder, the dataset is converted from a one-dimensional to a two-dimensional format. To provide an example, the number of inputs is transformed into a two-dimensional vector in order to increase the input dataset depending on timesteps. The vector is composed of two dimensions: one for the 10 timesteps and the other for the features (measured power data). This is shown as a vector with 10×1 dimensions. The encoder is currently processing the given input. Layer 1 is produced by the encoder module and is an LSTM network made up of ten LSTM cells. There is just one sample that each Cell unit processes for Memory that is stored for the long term as well as short-term memory (STM). The Long-Term Short-Term Memory (LSTM) system is comprised of a total of ten cells, each of which processes information in a manner that is sequential. The output generated by a single LSTM unit is sent to the subsequent LSTM unit in the sequence so that it may be processed further. The sample that was sent out by the first Long-Term Memory (LSTM) unit as the prior one is evaluated by the second LSTM unit, which determines whether or not it should be destroyed or kept. If the second LSTM determines that it is necessary to maintain the information, then it will be stored in the long-term memory. Furthermore, it will transmit both the processed feature information and the current sample to the third LSTM. Additionally, it will include the data layers from the first LSTM and the subsequent data from the third LSTM in the LSTM chain. The tenth and final Long-Term Storage-Short-Term Memory (LSTM) device in the system is responsible for receiving each of the nine samples that were previously processed by the LSTM cells that came before it model. All pertinent sample information is output by the final LSTM cell. The encoded characteristics are now represented by a 1 × 16 vector, which is the output that was produced. Note that in order to produce many copies of the 1 × 16 vector, a Repeat Vector was inserted as

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Layer 2, and the number of copies was proportional with regard to the number of discrete time steps. Consider, for example, how well the model handles the passage of time is set at ten time steps. As a result, Layer 2 produces ten duplicates of the encoded characteristics, creating a vector in two dimensions with the dimensions 10 by 16.

Step3: LSTM Decoder

The LSTM decoder's primary function is to carry out the duties of a sequence unfolding layer, which, after the completion of timestep sequence folding, brings back the original sequence structure of the input data to the form it is done before it was folded. This is done in order to reconstruct the outputs, and the manner in which the decoder interacts with the LSTM cells in order to rebuild the outputs is detailed in great length below in Figure 6.

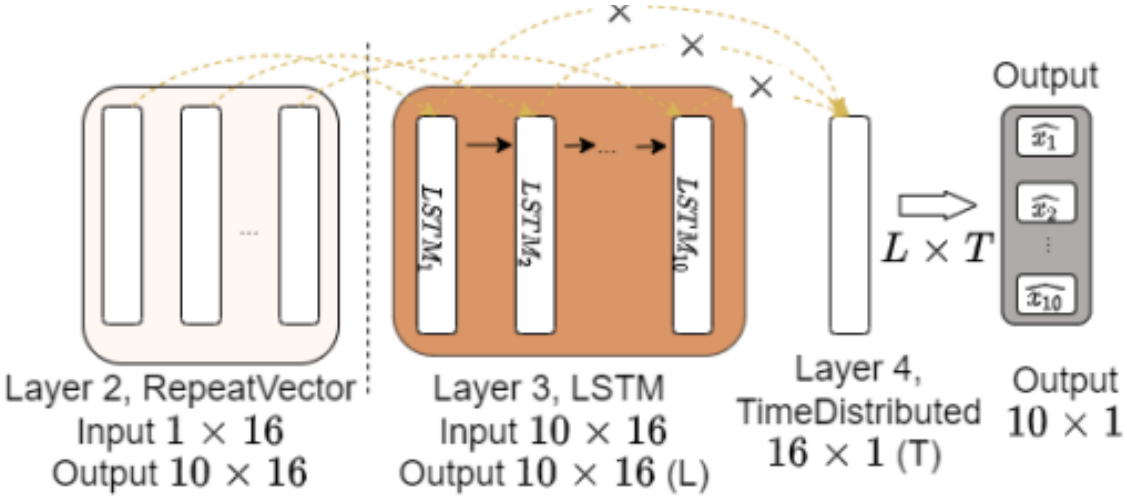

Figure 7 Details of LSTM Encoder

Each 1x16 set serves as the input for the decoder, and it utilises those sets to form a Layer 3 network that is made up of 10 LSTM cell units. The individual cell units that comprise a Long-Term Short-Term Memory (LSTM) are tasked with the responsibility of processing one-by-sixteen bit encoded characteristics of the data they store. The output that is produced by each Long Short-Term Memory (LSTM) unit represents the information that was gleaned from the encoded feature. This information may be thought of as a representation of the information. Following that, the end result of this output is multiplied by a 1 by 16 vector that was generated by the extra Time Distribution layer. Every LSTM cell unit concurrently generates a second output that stores the processed state of the current LSTM cell and passes it on to the next LSTM cell in the chain, except the very last LSTM unit. It is important to take note that matrix multiplication takes place both in the Time Distribution layer, which has dimensions of 16 1, and in the output of each LSTM layer (L), which has dimensions of 10 16. The output of this operation is a vector that has 10 dimensions and 1 dimension, which is same as the data dimensions.

Step4: Anomaly Detection

A popular definition of an anomaly is an observation that differs from the dominant trend that the bulk of the data exhibits. As a judgement point, a threshold may be set to indicate how far an observation deviates from the norm. Anomalous observations are those that go outside of the predetermined threshold. Using a dataset of measured power levels falling within a normal range, the model is trained using the threshold-based anomaly detection approach. I would want the reconstruction error rates for the Power data points that are typical. The maximum reconstruction error rate is determined as the threshold after training is finished and reconstruction errors for each sample have been calculated. Once the threshold has been established, which import measured power data from the testing dataset, which now contains a wide range of power values. For each measured power value, the reconstruction error rate is calculated for each individual sample that makes up the checking out the set. If the percentage of errors during rebuilding is greater than the threshold that was established, the sample is classified as being of an abnormal nature.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The method for determining the reconstruction loss for each sample in each time series sequence is shown in Figure 6. Let's say there are five different samples. [x1, x2, x3, x4, x5] Those are produced in the form of three different time-series sequences of [X1, X2, X3] with each sequence consisting of three samples that occur at three distinct points in time, where X1 [x1, x2, x3], X2 [x2, x3, x4], and X3 [x3, x4, x5] are represented by blue blocks with white dots. These three time-series of sequences serve as inputs for model, and the model produces outputs that correspond to each sequence individually X1 [x1, x2, x3], X2 [x2, x3, x4], and X3 [x3, x4, x5]. Let's assume that the original value for the 3 sequences are: X1 ∈ [x1 = 1, x2 = 2, x3 = 3], X2 ∈ [x2 = 2, x3 = 3, x4 = 4], and X3 ∈ [x3 = 3, x4 = 4, x5 = 5] where the mapping outputs for each time sequence came out as X^1 ∈ [^x1 = 1.1, x^2 = 2.02, x^3 = 3.01], X^ 2 ∈ [^x2 = 1.99, x^3 = 2.99, x^4 = 3.99], and X^3 ∈ [^x3 = 3.01, x^4 = 4.02, x^5 = 5.02]. For each sample, the reconstruction loss may be estimated as follows:

x1 = |1.1 – 1| / 1 = 0.1
x2 = (|2.01 – 2|+|1.97 – 2|) / 2 = 0.01
x3 = (|3.02 – 3|+|2.98 – 3|+|3.02 – 3|) / 3 = 0.01
x4 = (|3.98 – 4|+|4.02 – 4|) / 2 = 0.01
x5 = |5.01 – 5| / 1 = 0.02
In this particular scenario, the threshold that determines the maximum amount of reconstruction loss is set at 0.1. During the testing process, a sample is considered to have an anomaly if it has a reconstruction loss that is more than 0.1.

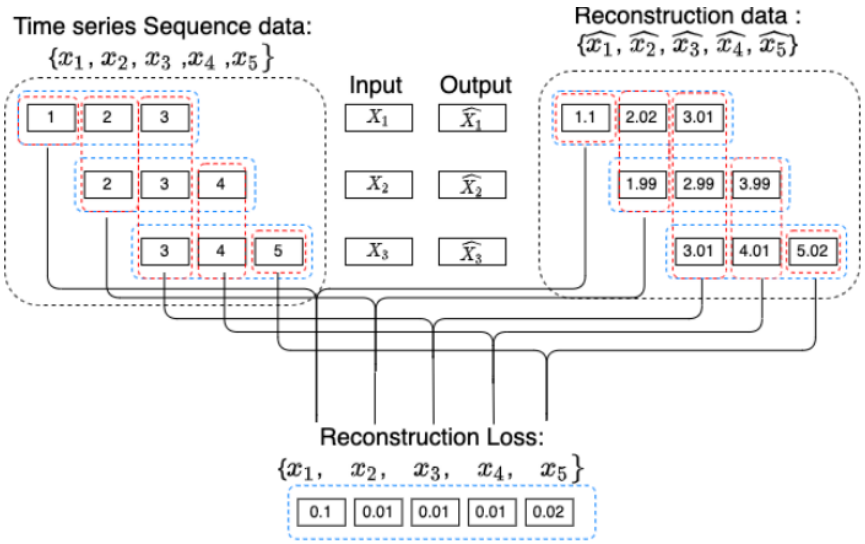

Figure 8 Computing Reconstruction Loss on Time Series

B. Algorithm

Algorithm 1 depicts the suggested model's algorithm. The training step in proposed approach has two key goals. To begin, the goal the goal of the training is to limit the amount of error that occurs during reconstruction such that the outputs that are rebuilt from the simplified representation of the input provide results that are almost identical to the original is practically possible. Second, the proposed determines the average percentage of incorrect reconstructions that are connected with the normal range. Points of data measurement in order to determine the best detection threshold to utilize during the test phase. The testing phase's primary purpose is to make use of the threshold in order to identify irregularities in the dataset being tested.

Training Phase

In Algorithm 1, specifically in Phase 1 titled "To sequence," the initial step of During the training phase, the initial dataset is restructured such that it is composed of time-series sequences. Within training dataset, each sequence, denoted as Xi, represents a collection of 10 measured data points across 10 timesteps.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Moving into the second phase, referred to as "LSTM-AE training inside Algorithm 1," the development of the model in question begins with the individual sequences being fed into the encoder one at a time. Every data point is handled in an orderly fashion by a single LSTM unit throughout the processing of each sequence. Following the completion of the training for a particular sequence, the encoder's latent space will translate the important data points into a one-dimensional representation of the encoded characteristic, which is then duplicated multiple times by the Repeat Vector layer.

The decoder is constructed with an LSTM network in which the number of LSTM cells is determined by the number of timesteps that correspond with the copies of the encoded feature. One LSTM cell is used to decode and process each individual feature, and the output of this cell is used to Time Distributed Dense Layer results in a single-dimensional vector containing the outcomes of all LSTM cell operations.

In the subsequent stages, particularly from stages 8 to 13, a reconstruction loss is calculated by comparing the data that was entered to the decoder with the data that was produced. A backpropagation strategy is used in order to fine-tune the model's weights and parameters. As was previously stated, we use a method known as Mean Absolute Error, or MAE, as the function for the loss of reconstruction errors in Equation 12.

$$Loss(MAE) = \frac{\sum_{i=1}^{n} \left| x_i - x_i \right|}{n} \qquad (12)$$

where n is the total number of samples, xi is the input that was originally supplied to the encoder, and xi is the output that was created by the decoder. where xi is the output that was produced by the decoder.

The model undergoes training on each time-series sequence until the reconstruction loss for each sample reaches a satisfactory threshold. Notably, the activation function denoted by "*tanh*" is implemented in the model that propose to use, and the output of the tenth LSTM is encoded in the latent space of the encoder. Furthermore, we employ a rate of 0.2 for the Dropout layers of both the encoder and the decoder, for a grand total of two tiers. A further Time Distributed Dense layer is appended just prior to the output layer. The Repeat Vector layer is positioned in the midsection, between the encoder and the decoder. Following the conclusion of the training process, the maximal reconstruction error will be computed and subsequently employed as a threshold. Further the development of this concept will occur during Phase 3, which involves adjusting the threshold.

Testing Phase

In Phase 4, "Anomaly detection on testing set," we outline the specific procedures for testing the model. During this phase, the trained LSTM encoder processes a sequence of time points called a time series that consists of 10 data points, each of which is separated by 10 timesteps. It is important to note that these 10 data points cover the complete spectrum of power values.

Using the feature representation generated by the LSTM encoder, which The LSTM decoder, which simultaneously decreases the dimensionality of the input sample and generates a single time series that has ten data items and ten timesteps. For each data point in this sequence, we compute the reconstruction error rate, following the same strategy as discussed in Equation 12 during the training phase. If the reconstruction loss for a data point exceeds a predefined threshold, it is classified as an unusual occurrence; otherwise, it is considered to be the standard. The schematic of this categorization procedure is depicted in Equation 13. $X' = \begin{cases} X_i' \text{ is Anomalies}, & if ltest_{arr}[i] > \eta \\ X_i' \text{ is normal}, & otherwise \end{cases}$ (13)

where X' denotes a reconstructed time-series, X_i' denotes a data point inside the time-series, and $if ltest_{arr}[i]$ is the outcome of a reconstruction loss function using MAE.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Algorithm 1 LSTM-AE Anomaly Detection

Input:

Training set $\{x_0, x_1, x_2, \ldots, x_{n-1}\}$,

Test set $\{x'_0, x'_1, x'_2, \ldots, x'_{m-1}\}$,

Timesteps t

Output: A Set of anomalies(A_t) or normal (N_t)

Begin

/ Phase 1: To sequence */*

X' X'_i : sets of training and testing data based on timesteps (t=10)

for i ∈ [0, n – t) **do**

$X_i = [x_i :: x_{i+t}]$

End

for i ∈ [0, m – t) **do**

$X'_i = [x'_i :: x'_{i+t}]$

End

/ Phase 2: LSTM-AE training */*

Initialize the parameter of LSTM-AE model (M)

for $Xi \in [X_0, X_1, \ldots, X_{n-t})$ **do**

$\hat{X}_i = M(X_i)$

$L_{err} = \sum |Xi - \hat{X} i|$

Update LSTM-AE to minimize L_{err} by Eq. 9

End

/* Phase 3: Threshold setting */

Function RLOSS(X):

/* X_i reconstruction error calculation */;

for i ∈ (0, n – t) **do**

$\hat{X} i = M(X_i)$

$Err_{arr}[i, i : i + t] = |\hat{X}_i - X_i|$

End

return Err_{arr};

End Function

/* All data reconstruction error calculation */

for i ∈ (0, n) **do**

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$l_{arr}[i] = \sum Err_{arr}[:, i] / \sum (Err_{arr}[:, i]! = 0)$

end

return *larr*

/* Max RLoss from training set */

$threshold(\eta) = max(RLOSS([\ X'_{0,} X'_{1,} \ldots\ldots X'_{n-t,}\ ])$

/* Phase 4: Anomaly detection on testing set */

$ltest_{arr} = RLOSS([\ X'_{0,} X'_{1,} \ldots\ldots X'_{m-t,}\ ])$

for i ∈ (0, m) **do**

if $ltest_{arr}[i] > \eta$ **then**

$x'_i \rightarrow A_t$

else

$x'_i \rightarrow N_t$

end

end

end

Table 1 Confusion Matrix

Total Population		Predicted Condition	
		Normal	Anomaly
Actual Condition	Normal	TN	FP
	Anomaly	FN	TP

- True Positive (TP) signifies the correct classification of an anomalous data point as anomalous.
- True Negative (TN) denotes the accurate classification of a piece of data that is typical as usual.
- False Positive (FP) shows that a typical data point was mistakenly identified as an unusual one.
- False Negative (FN) represents the incorrect categorization of an abnormal data point as typical.

In accordance with the aforementioned criteria, the evaluation metrics are computed as follows:

$$RECALL = \frac{TP}{TP + FN} \quad (15)$$

$$PRECISION = \frac{TP}{TP + FP} \quad (16)$$

$$F1 - score = 2 \times \left(\frac{PRECISION \times RECALL}{PRECISION + RECALL} \right) \quad (17)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (18)$$

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

I. RESULTS

The simulation of the proposed system, as depicted in Figure 4, was carried out utilising MATLAB-SIMULINK 2021a version. The measurements of power consumption utilised in this simulation are acquired through a channel that incorporates additive white Gaussian noise (AWGN) and has a twenty-decibel signal-to-noise ratio (SINR). After the power consumption measurements have been gathered, they are employed in the training process of a Long Short-Term Memory (LSTM) autoencoder, which falls under the category of neural networks.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 48, 1)]	0
lstm (LSTM)	(None, 48, 50)	10400
dropout (Dropout)	(None, 48, 50)	0
lstm_1 (LSTM)	(None, 50)	20200
dropout_1 (Dropout)	(None, 50)	0
dense (Dense)	(None, 1)	51
activation (Activation)	(None, 1)	0

Total params: 30651 (119.73 KB)
Trainable params: 30651 (119.73 KB)
Non-trainable params: 0 (0.00 Byte)

Figure 9 modelled architecture of the system

A single output 1 and a sequence input value of 48 are included in the modelled architecture of the system that is executed in Figure 9. After that, a training set and a validation set are constructed from the dataset after it is partitioned in an 80:20 ratio. Both the training loss and the validation loss for the autoencoder model are shown in Figure 10 and Figure 11, respectively. The numbers provide an illustration of both of these monetary losses. The loss curves provide very useful and insightful perspectives on the performance of the model during the whole of the training and validation processes stages. A model's ability to learn from training data is shown by its training loss, while its ability to generalize to new data is indicated by its validation loss.

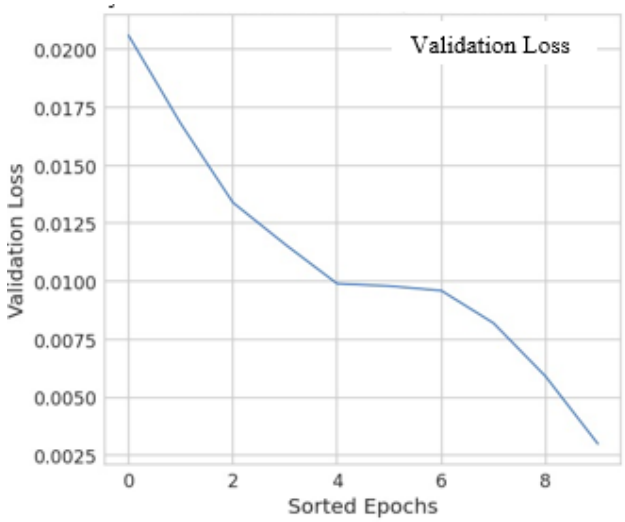

Figure 10 Validation Loss over Epochs

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

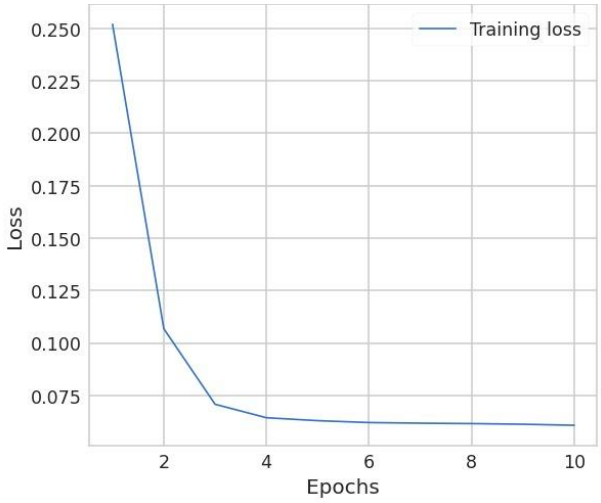

Figure 11 Training Loss graph over Epochs

When the process of training has been finished, the model is applied to the task of making predictions about the behaviour of the signal within a designated timeframe. The mean absolute errors derived from these predictions are subsequently utilised to calculate thresholds for anomaly detection. These thresholds aid in the identification of deviations or deviations within the signal. Figure 12 shows the density of the calculated errors, there by presenting a graphical representation of the distribution of prediction errors. The examination of this density plot allows an improved understanding of the fluctuations in the model's predictions and helps in the development of accurate standards for identifying anomalies in the signal data.

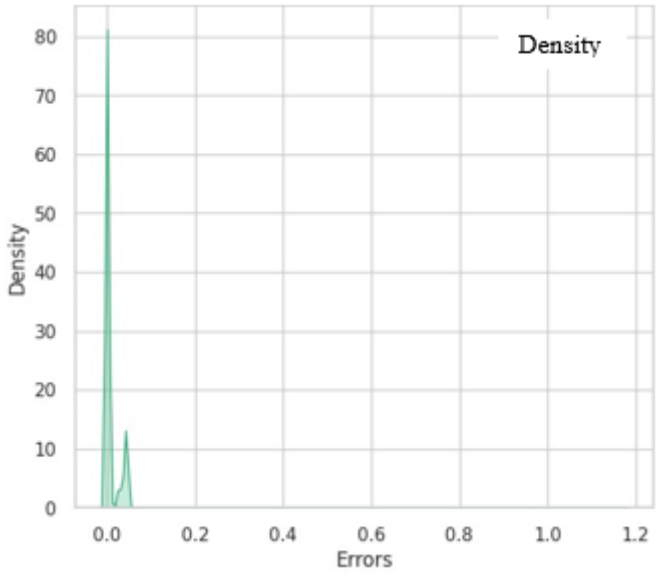

Figure 12 Kernel Density Estimation Plot of Errors

The trained autoencoder model's performance is assessed via the process of computing True Positives (TP), True Negatives (TN), and False Positives (FP) across a number of different threshold values. the procedure also includes calculating both false positives (FP) and false negatives (FN). These assessment metrics give information about the model's capacity to recognises anomalies and non-anomalies in data. In Table-2, details the fluctuations of these parameters for various threshold values, providing a full picture of the model's performance under varied sensitivity levels. Figure 13 depicts how these indicators vary across multiple threshold settings, assisting in the selection of an optimal threshold for successful anomaly identification.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 2 Parameters of Confusion matrix

Threshold	TP	TN	FP	FN
0.02	28670	119596	4	3
0.04	28668	119595	6	4
0.07	28666	119592	8	7
0.09	28665	119591	9	8
0.11	28662	119588	12	11
0.13	28658	119587	16	12
0.16	28654	119585	20	14
0.18	28650	119581	24	18
0.2	28649	119423	25	176

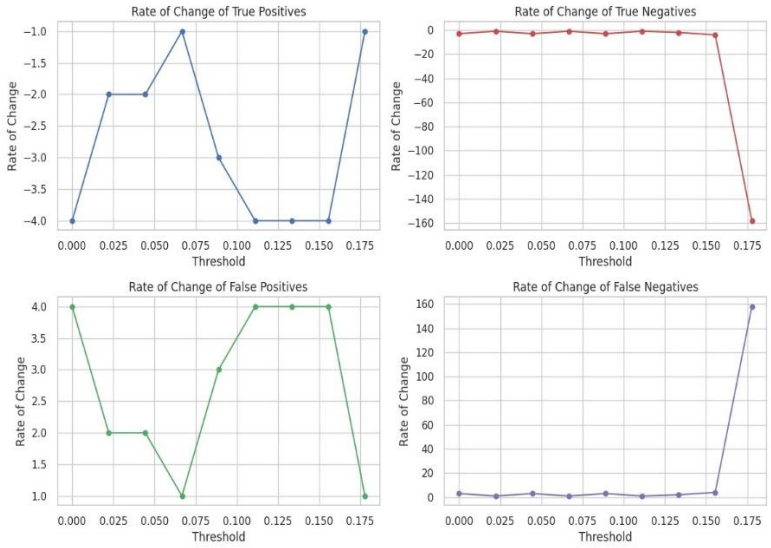

Figure 13 multiple threshold settings

Later on, these parameters (TP,TN,FP,FN) are utilized to calculate the accuracy, precision ,Recall, F1-score Table 3 using the equations (14 to 18).

Table 3 Values of performance metrics

Threshold	Accuracy	Precision	Recall	F1-Score
0.02	0.99995279	0.99995279	0.99995279	0.999978

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

0.04	0.99993 2557	0.999790 751	0.999 86	0.9998 26
0.07	0.99989 8835	0.999721 002	0.9997 56	0.9997 38
0.09	0.99988 5347	0.999686 127	0.9997 21	0.9997 04
0.11	0.99984 4881	0.999581 502	0.9996 16	0.9995 99
0.13	0.99981 1159	0.999442 003	0.9995 81	0.99951 2
0.16	0.99977 0693	0.999302 504	0.9995 12	0.9994 07
0.18	0.99971 6739	0.999163 005	0.9993 72	0.9992 68
0.2	0.99864 4392	0.999128 13	0.9938 94	0.9965 04

The above mentioned table emphasizes the vital relevance of choosing the right threshold value when it comes to attaining a high-performance measure for the autoencoder in detecting theft. The threshold value chosen has a major impact on the model's capacity to reliably detect cases of theft and non-theft. It emphasizes the importance of fine-tuning the threshold value in optimizing the model's performance and increasing its efficacy in identifying anomalies or theft-related trends in data. The threshold is an important parameter that may be set to create a compromise between sensitivity and specificity, ensuring that the model can detect possible theft while minimizing false alarms

DISCUSSION

In this section, the findings of main study underscore the efficacy of the proposed system, which underwent comprehensive simulation using MATLAB-SIMULINK. The analysis encompassed a comprehensive evaluation of power consumption measurements acquired via an Additive White Gaussian Noise (AWGN) channel, accurately simulating noisy real-world scenarios with a Signal-to-Noise Ratio (SINR) of 20 decibels (20dB). The aforementioned measurements are utilised in the process of training a specific neural network that is called an autoencoder for the Long-Term Short-Term Memory (LSTM). The data can be properly captured despite their complicated patterns thanks to the capabilities of this particular network. The model is subjected to a comprehensive test to see how well it could identify abnormalities and identify instances of possible theft. Metrics of performance such as the F1 score, True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), along with Accuracy, Precision, and Recall are all considered all exhibited near-perfect values.

The results of this study emphasize the importance of carefully choosing a suitable threshold value in order to acquire the best possible performance from the autoencoder when it comes to the prevention of theft. In order to strike a healthy equilibrium between sensitivity and robustness, it is very necessary to fine-tune this threshold with extreme care specificity. This will guarantee accurate identification of potential theft while minimizing the occurrence of false alarms. The comprehensive analysis presented in this study are the effectiveness of the system and its potential for real-world applications in the domains of theft detection and anomaly identification within complex systems. This is achieved through a combination of simulation, model training, anomaly detection, and performance evaluation. This study provides valuable insights into the development of robust anomaly detection models, which have broad practical applications across diverse domains.

.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

REFRENCES

[1] S. Lee *et al.*, "Smart Metering System Capable of Anomaly Detection by Bi-directional LSTM Autoencoder," *Dig. Tech. Pap. - IEEE Int. Conf. Consum. Electron.*, vol. 2022-January, 2022, doi: 10.1109/ICCE53296.2022.9730398.

[2] C. Chahla, H. Snoussi, L. Merghem, and M. Esseghir, "A novel approach for anomaly detection in power consumption data," *ICPRAM 2019 - Proc. 8th Int. Conf. Pattern Recognit. Appl. Methods*, no. Icpram, pp. 483–490, 2019, doi: 10.5220/0007361704830490.

[3] C. W. Tsai, K. C. Chiang, H. Y. Hsieh, C. W. Yang, J. Lin, and Y. C. Chang, "Feature Extraction of Anomaly Electricity Usage Behavior in Residence Using Autoencoder," *Electron.*, vol. 11, no. 9, 2022, doi: 10.3390/electronics11091450.

[4] O. I. P. Y. M. L. M. M. Veres, "Unsupervised Anomaly Detection in Time Series Using LSTM-Based Autoencoders".

[5] M. Adil, N. Javaid, U. Qasim, I. Ullah, M. Shafiq, and J. G. Choi, "LSTM and bat-based rusboost approach for electricity theft detection," *Appl. Sci.*, vol. 10, no. 12, pp. 1–21, 2020, doi: 10.3390/app10124378.

[6] C. Chahla, H. Snoussi, L. Merghem, and M. Esseghir, "A deep learning approach for anomaly detection and prediction in power consumption data," *Energy Effic.*, vol. 13, no. 8, pp. 1633–1651, 2020, doi: 10.1007/s12053-020-09884-2.

[7] M. Nazmul Hasan, R. N. Toma, A. Al Nahid, M. M. Manjurul Islam, and J. M. Kim, "Electricity theft detection in smart grid systems: A CNN-LSTM based approach," *Energies*, vol. 12, no. 17, pp. 1–18, 2019, doi: 10.3390/en12173310.

[8] S. Parsai and S. Mahajan, "Anomaly Detection Using Long Short-Term Memory," *Proc. Int. Conf. Electron. Sustain. Commun. Syst. ICESC 2020*, no. Icesc, pp. 333–337, 2020, doi: 10.1109/ICESC48915.2020.9155897.

[9] S. C. Yip, W. N. Tan, C. K. Tan, M. T. Gan, and K. S. Wong, "An anomaly detection framework for identifying energy theft and defective meters in smart grids," *Int. J. Electr. Power Energy Syst.*, vol. 101, no. February, pp. 189–203, 2018, doi: 10.1016/j.ijepes.2018.03.025.

[10] S. Hussain, M. W. Mustafa, T. A. Jumani, S. K. Baloch, and M. S. Saeed, "A novel unsupervised feature-based approach for electricity theft detection using robust PCA and outlier removal clustering algorithm," *Int. Trans. Electr. Energy Syst.*, vol. 30, no. 11, 2020, doi: 10.1002/2050-7038.12572.

[11] C. H. Park and T. Kim, "Energy theft detection in advanced metering infrastructure based on anomaly pattern detection," *Energies*, vol. 13, no. 15, 2020, doi: 10.3390/en13153832.

[12] H. Barzamini and M. Ghassemian, "Comparison analysis of electricity theft detection methods for advanced metering infrastructure in smart grid," *Int. J. Electron. Secur. Digit. Forensics*, vol. 11, no. 3, pp. 265–280, 2019, doi: 10.1504/IJESDF.2019.100475.

[13] H. S. Nam, Y. K. Jeong, and J. W. Park, "An Anomaly Detection Scheme based on LSTM Autoencoder for Energy Management," *Int. Conf. ICT Converg.*, vol. 2020-October, pp. 1445–1447, 2020, doi: 10.1109/ICTC49870.2020.9289226.

[14] C. Kaymakci, S. Wenninger, and A. Sauer, "Energy Anomaly Detection in Industrial Applications with Long Short-term Memory-based Autoencoders," *Procedia CIRP*, vol. 104, no. March, pp. 182–187, 2021, doi: 10.1016/j.procir.2021.11.031.

[15] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Deep Autoencoder-based Detection of Electricity Stealth Cyberattacks in AMI Networks," *ISSCS 2021 - Int. Symp. Signals, Circuits Syst.*, no. July, 2021, doi: 10.1109/ISSCS52333.2021.9497376.

[16] M. Kardi, T. AlSkaif, B. Tekinerdogan, and J. P. S. Catalão, "Anomaly Detection in Electricity Consumption Data using Deep Learning," *21st IEEE Int. Conf. Environ. Electr. Eng. 2021 5th IEEE Ind. Commer. Power Syst. Eur. EEEIC / I CPS Eur. 2021 - Proc.*, pp. 1–6, 2021, doi: 10.1109/EEEIC/ICPSEurope51590.2021.9584650.

[17] M. Farahani, "Anomaly Detection on Gas Turbine Time-series Data Using Deep LSTM-Autoencoder," 2020.