Synergy: Cross-Disciplinary Journal of Digital Investigation
(ISSN 2995-4827) VOLUME 03 ISSUE 04, 2025

DEEP LEARNING TECHNIQUES FOR TIME SERIES DATA MINING ANOMALY DETECTION

Media Noaman Solagh
University of Baghdad, Baghdad, IRAQ

Abstract:

As communications innovations develop and ventures become automated, a wide scope of applications and frameworks have arisen to provide and produce enormous measures of data. Numerous techniques have been proposed to separate key indicators from a lot of data to address the state of the whole framework. Inconsistencies utilizing such indicators are identified quickly to forestall likely accidents and limit economic misfortunes. Multivariate anomaly detection of time series data is especially difficult because it requires concurrent consideration of time, dependencies, and relationships between factors. Profound learning-based works have made tremendous advancements around here. Representations of huge scope successions in records may be prepared and advanced in an unaided way and identify oddities from the data. Be that as it may, the vast majority of them are unmistakable to an individual use case and in this manner require space information for legitimate arrangement. This paper provides a logical foundation for anomaly detection in time series data and surveys state-of-the-workmanship certifiable applications. We likewise investigate techniques appropriate for profound time series anomaly detection models utilizing a few standard datasets. At long last, we present a plan for choosing and preparing a fitting model, a profound learning-based time series anomaly detection procedure.

Keywords: Deep Learning Techniques, Multivariate Anomaly Detection, Communications Technologies, Recursive Neural Networks (RNNs), Long-Short Term Memory (LSTM).

1. Introduction

Anomaly detection is the process of identifying data points or patterns in a data set that deviate significantly from the norm. A time series is a collection of data points collected over some time [1][2].Detecting anomalies is one of the main challenges to ensure security in private wireless networks. WSNs are exposed to various threats that may generally cause data corruption and produce false measurements. Detection of such anomalous data is required to reduce false alarms. Anomaly detection examines specific data points and detects rare events that appear suspicious

because they differ from the approved pattern of behaviors [3][4]. Detecting bugs is nothing new, but as data increases, manual tracking becomes impractical. Anomalies in time series data mining WSNs refer to any unusual, abnormal, or unexpected behavior or events in the flow of collected sensor data that differ from expected or normal patterns. Anomaly detection is also a relevant problem in the field of data analysis. In networked systems, where single entities interact in pairs, anomalies are observed when the pattern of interactions deviates from patterns considered regular [5][6]. Figure 1 shows an illustrative diagram of the anomaly types in time series data samples [1].

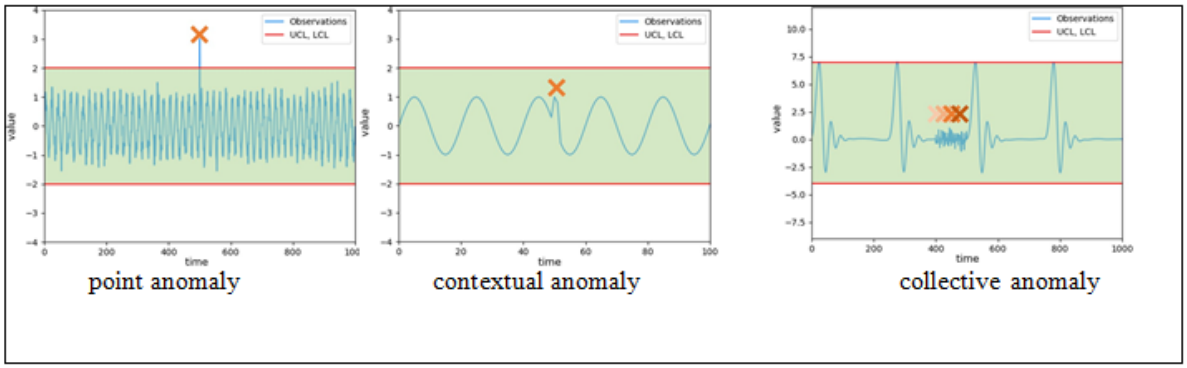


Figure 1: Anomaly types in time series data samples [1].

We note from Figure 1 above that the process of anomaly detection is the process of identifying unexpected things, where elements or events from the data are an area of interest. For many researchers and practitioners, he is now one of the main tasks in data mining and quality assurance [4][7]. We might observe the types of anomaly detection in a variety of application areas and forms of temporal data, which takes multiple faces and forms. Researchers have presented a range of classical anomaly detection methods including techniques based on linear models, distance-based methods, density-based methods, and support vector machines, it is still a viable option for the algorithm. Anyhow, as a goal systems become larger and more complex with these methods [1][8]. A schematic diagram of the anomaly detection operation with the scope of every anomaly kind in WSNsis presented in Figure 2 [7].

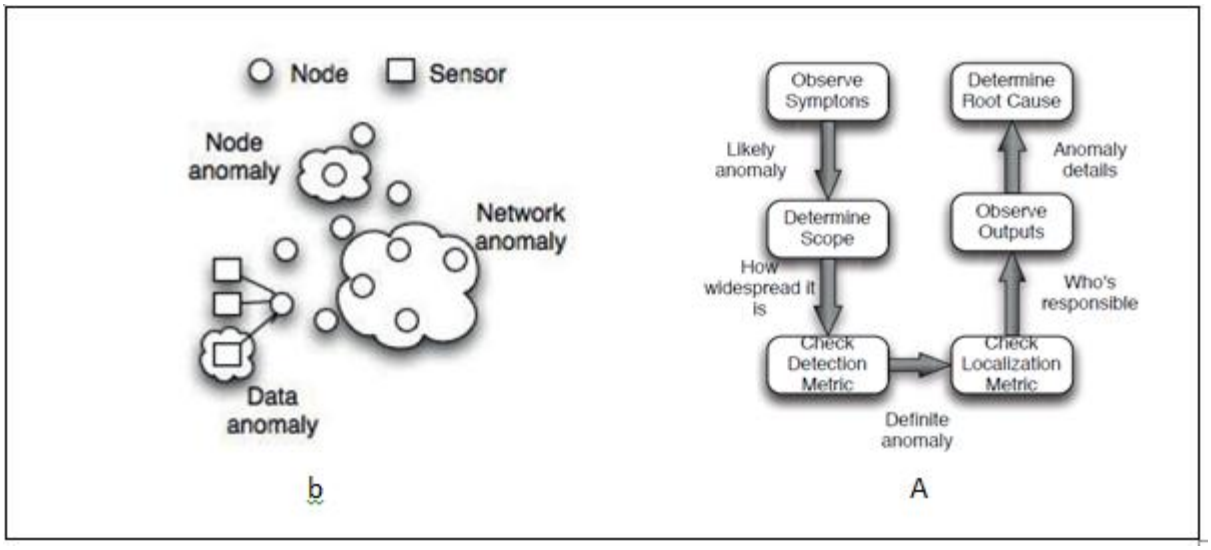


Figure 2: Diagram of the anomaly detection operation in WSNs, (a) Flow chart structure, (b) Demonstration of every anomaly kind [7].

By concerning the flow chart presented in Figure 2, we might observe the process of the anomaly detection activity flow in the WSNs. Conditions indicating an anomaly related to the client's policy. For example, the operator specifies the frequency and period required for the data to be delivered by

the sensor nodes. Such set detection thresholds and reporting anomalies to operators. So that whenever the data through the sensor node is variable according to the expected schedule, the operator might be notified. Like that frequency, urgency, and level of detail might also be regarded within a notification definition of the client. Anomaly diagnostics was created in the management of traditional network equipment [7], however, it is revisited here along the perspective of WSN to clarify how it does so it might be employed for various kinds of WSN anomalies.

2. Related work

2.1 Time Series Anomaly Detection

This exhaustive logical review assesses the most recent painstakingly planned anomaly detection algorithms. Where 71 anomaly detection algorithms were arranged and yet again carried out from different source areas and assessed on 976-time series datasets. Algorithms are considered browsed different algorithm families and detection strategies to address the full scope of anomaly detection techniques. In this article, we provide a concise outline of the innovations and their commonalities. We assess qualities and shortcomings and subsequently consider factors like viability, efficiency, and strength. Our exploratory aftereffects of this study ought to reduce the issue of algorithm selection and open up new examination directions [9]. Figure 3 shows a large portion of the strategies utilized for anomaly detection for time series data organized by their family technique [10].

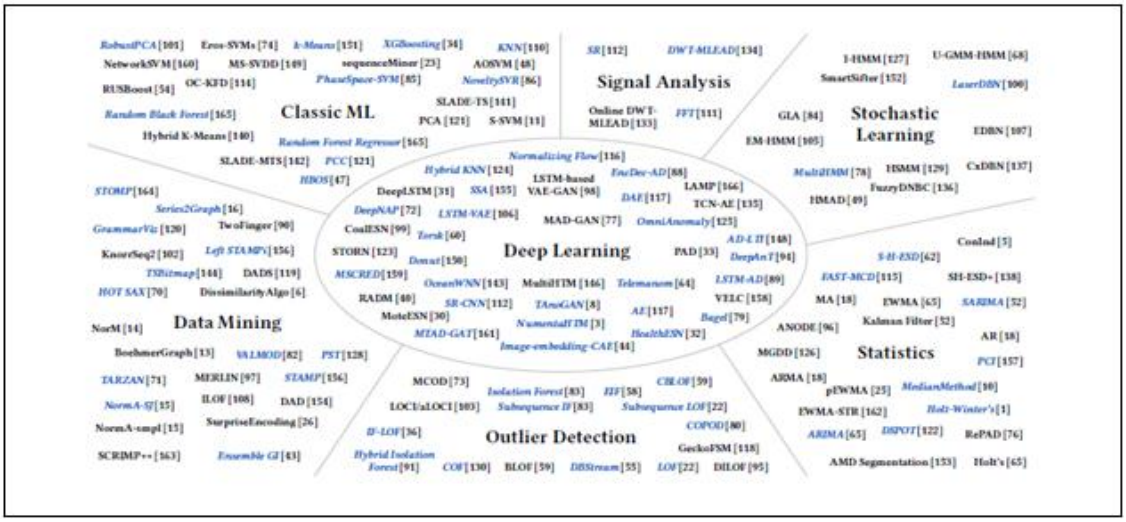


Figure 3: The complete 158 anomaly detection approaches for time series data arranged by their family method [10].

All 158 anomaly detection techniques were combined for the time series data and formed by their family approach. The techniques regarded in such evaluation are highlighted with blue italics.

2.2 Anomaly Detection with Deep Approaches

In information with complex designs, deep neural networks are strong techniques for demonstrating conditions. Various researchers have explored their application to anomaly identification utilizing an assortment of deep learning models, as delineated in Figure 4. In this study, we focused on the anomaly detection of the time series data utilizing Deep Learning techniques. Figure 4 shows the architecture utilized in the time series anomaly detection.

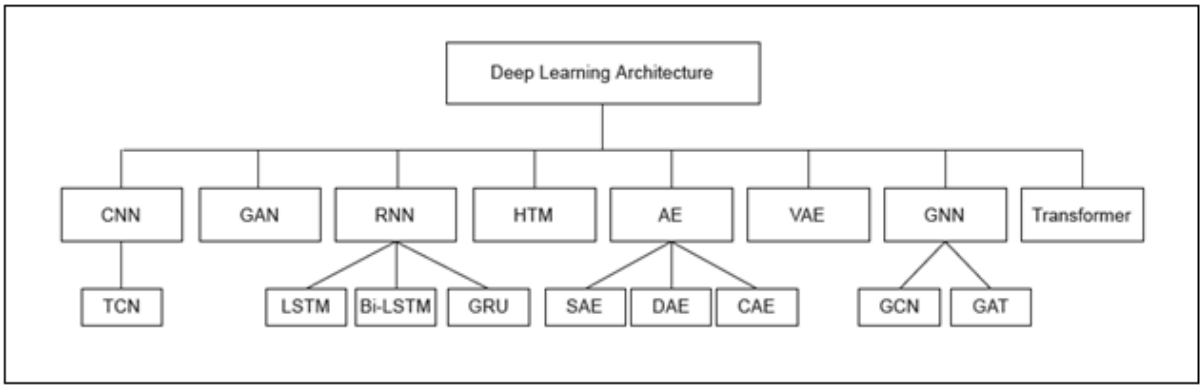


Figure 4: The architecture utilized for the time series anomaly detection [12].

In our review, deep anomaly models were utilized, and time series identification was arranged in view of its methodology and principal structure. There are two primary ways of learning part in Figure 5, in the writing on time series anomaly identification prediction-based and remaking-based. The prediction-based model may be set up to predict the next timestamp, while the replay-based model might be set up to generate [12].

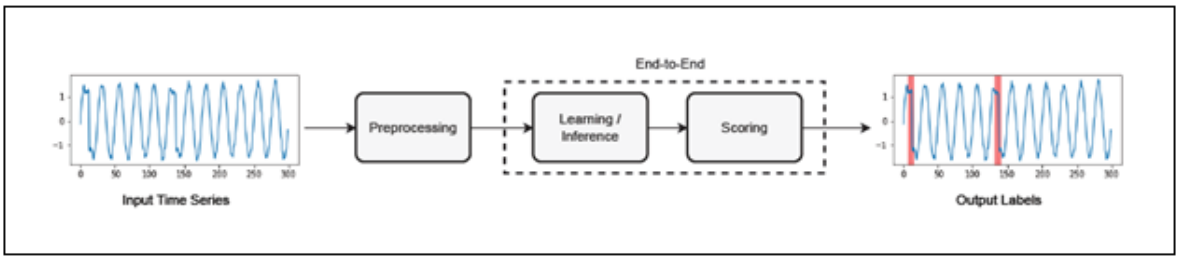


Figure 5: General parts of deep time series anomaly discovery schemes [12].

Figure 5 displays the classification of profound learning architectures in time series anomaly detection. This study summarizes the time series anomaly detection models based on the information dimensions processes, which are invariant and multivariate time series.

3. Methodology

The most important methods for removing anomalies in time series data are forecasting methods, reconstruction-based models and hybrid methods [11][13]. In this research, the focus will be on forecasting methods due to their importance and the fact that they depend on forecasting processes and are represented by a moderate and uncomplicated structure in addition to high efficiency in implementation and training and not consuming a lot of time or many calculations [11].

3.1 Forecasting-based models

The prediction-based approach uses a learned model to predict a point or subsequent sequence based on a point or event window. To determine how unusual the incoming values are, the expected values are compared to their real amounts. Their deviations from the real amounts are considered outliers. Most forecasting approaches utilize a sliding window to forecast a single point at a time. To identify unusual behavior, they use a predictor to model standard behavior. This is especially useful in real-world abnormality detection cases where behavior is typically plentiful, yet strange behavior is rare [12][14]. The forecasting-based techniques will be listed and discussed in this section. These techniques are the recurrent neural networks (RNNs), the convolutional neural networks (CNNs), and the graph neural networks (GNNs), which will be discussed in the upcoming paragraphs in detail.

3.1.1 Recurrent Neural Network (RNN).

One of the first studies on detecting fake images using machine learning was conducted by researchers; those suggested a statistical approach for digital picture forgery recognition by analyzing discrepancies in the picture characteristics. Since then, various machine learning-based techniques have been developed to reconstruct fake pictures, containing deep learning schemes such as recurrent neural networks (RNNs). A feed-forward brain network having at least single stowed-away layers with something like one feedback circle is referred to as an intermittent network as displayed in Figure 6 [8].

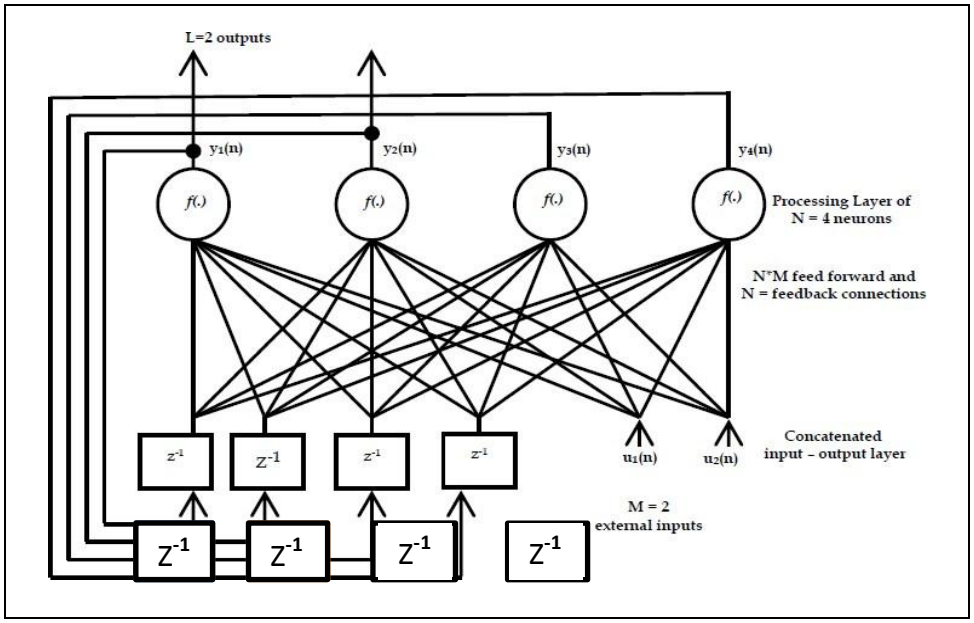


Figure 6: Schematic diagram of Recurrent Neural Networks, RNN [8].

As shown in Figure 6, the feedback may be autonomic, that is, the result of the action of the neuron is determined by its method of training. The feedback circuits involve the use of delay unit components with several regions, leading to a non-linear dynamic path of behavior, as a smart grid would be expected to have indirect units. Different alternative types may differ in the method of internal linking, but they achieve the same goal and desired result, which is the idea of applying repetition [6][13].

Since RNNs have internal memory, they might process entered sequences of variable length and exhibit temporal dynamics. An example of a simple RNN structure could be observed in Figure 7 [5].

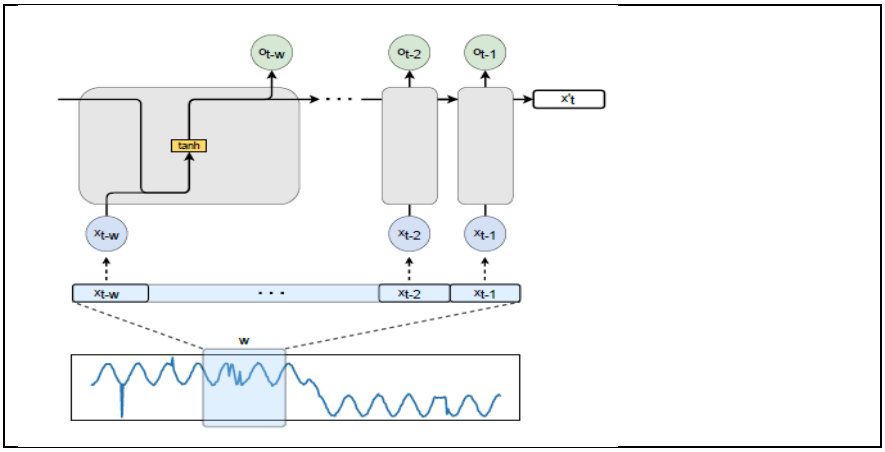


Figure 7: An example of a simple RNN structure [5].

Concerning the structure of the RNN algorithm presented in Figure 7, we could notice the recurring units which will take input points window $X_{t-w:t-1}$ with predicts the new timestamp as output, $x'.t$. Recursively, the entered sequence is fed to the network timestamp by timestamp. Applying the entered sequence x_{t-1} of the recurrent unit o_{t-2}, with the activation function as $tanh$, the resulting vector x'_t is computed utilizing the below relations:

$$x'_t = \sigma(W_{x'}, o_{t-1} + b_{x'}),$$

$$o_{t-1} = \tanh(W_{o.x_{t-1}} + U_o.o_{t-2} + b_h) \quad (1)$$

Whereas Wx', Wo, Uo, and b are the network components. Redundancy occurs when the network utilizes previous results as entered to remember what one learned along the past steps. This is where the network learns the long- and short-term expectations [4][8]. RNNs techniques include three main algorithms types, which are the Recurrent neural network (RNN) algorithm, the Long-short-term memory (LSTM) algorithm and the Gated recurrent unit (GRU) algorithm as shown in Figure 8.

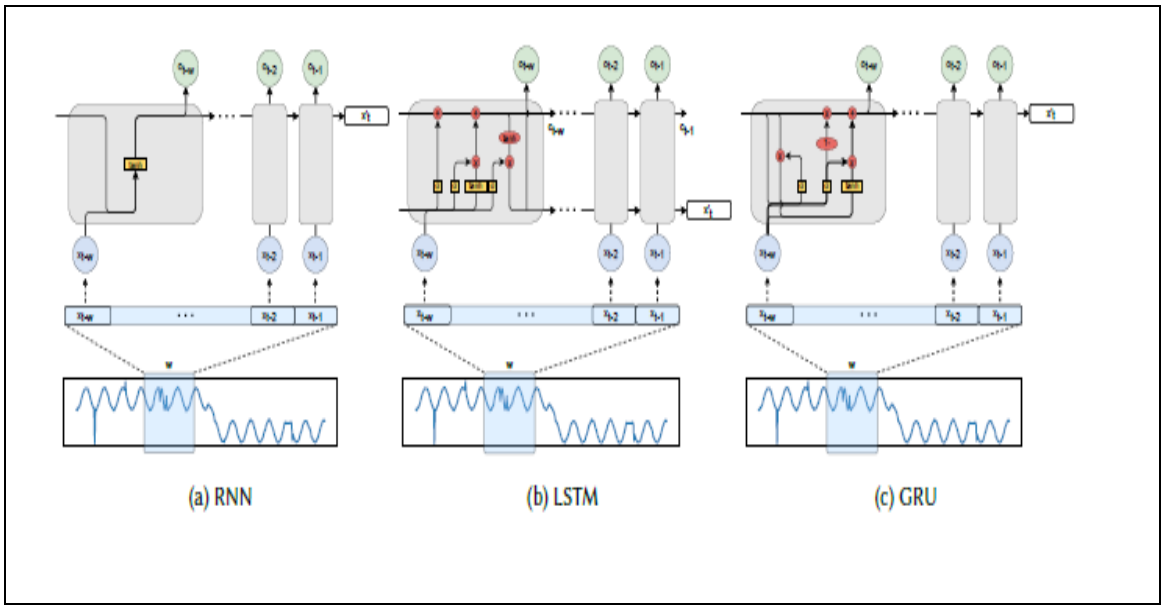


Figure 8: Forecasting-based models overview, (a) recurrent neural network (RNN), (b) long short-term memory module (LSTM), and (c) gated recurrent module (GRU) [7].

3.1.2 Deep Learning Generative Adversarial Networks (GANs)

authors have conducted further studies on detecting fake images utilizing machine learning and recommended modern, statistical techniques to identify digital image forgeries by analyzing discrepancies in picture characteristics. Since then, various machine learning-based techniques have been advanced to reconstruct fake images, those include deep learning schemes such as the Generative Adversarial Networks (GANs). Generative Adversarial Networks (GANs) are preset and they rely on the concept of affording two networks that compete against each other. One network is in charge of the generation of fake training data looking real, along with learning how to approximate the distribution that generated the real data Training data. Such a network is termed a generator, represented by G(z) because it holds a vector of random noise z as entrance, and assigns it to a generated data point (the picture, in our issue). The other network, named discriminator, or critic, is utilized to differentiate between generated and real samples. It is termed D(x) whereas the network takes a sample x and results in the probability that x is distinct from the real data set. It could be a competition between the two networks It is described as a min-max game between two players trying to beat each other, as shown in Figure 9 [8].

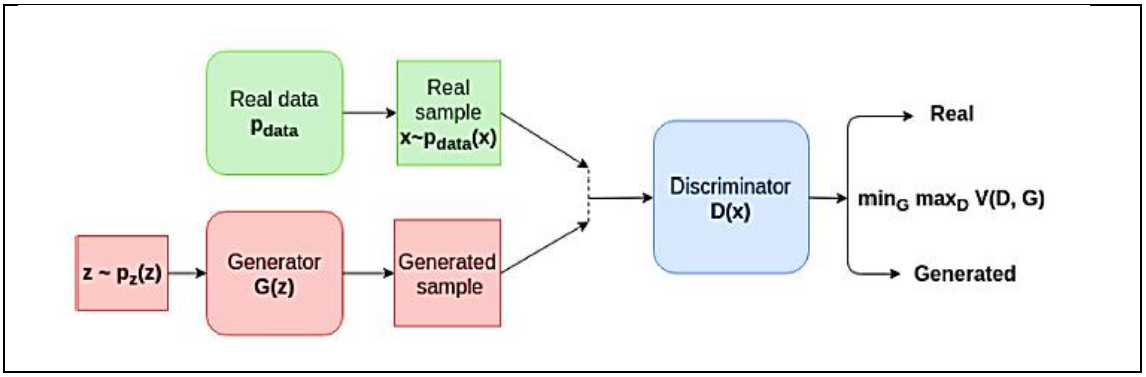


Figure 9: An architecture of typical GAN with loss function [8].

Referring to Figure 9, the construction of the GANs with multiple convolutional layers is illustrated called deep convolutional GANs (DCGANs) and it has employed among other approaches to generate realistic pictures, image noise minimization, image translation, and picture completion. Image completion is often accomplished utilizing architectural generators identical to autoencoders, which promotes learning the latent representation of information. Latent data representation is achieved by compression by decompressing the entered info in a form that minimizes data loss [8][13]. While the minimum-maximum objective that directs the training of GANs is theoretically easily put, training GANs to provide valid outcomes is a hard mission. The aim of generative modeling is basically to compose the fake data distribution that the generator learns to sample, *Pg*, as identical as possible to distribute real data public relations. Anyhow, if one tries to do this utilizing the joint distribution spacing/distance metrics, such as the Kullback-Leibler (KL) divergence, which are commonly utilized to train GANs, enhancing the procedure is often hard in practice because problems such as discontinuities and/or fading within gradients an objective function concerning the network components, that might develop whenever the true sample is not within public relations support of P_r. The Wasserstein distance, described shortly, has been proposed as a solution to such problems. Also, it is an essential parameter of the loss function in our structure. The Wasserstein extension amidst two distributions might be noticed intuitively as the minimum attempt needed to travel the probability mass among such distributions, which is theoretically expressed as:

$$W(P_r, P_g) = inf_{y \in \prod(P_r, P_g)} E_{(x,y)\sim y}[\|x - y\|] \qquad (2)$$

Such that *Pr*, *Pg* is the set of each distribution whose marginal distributions are Pr and *Pg*. Such relation is not surprisingly practically intractable, yet a further useful expression might be achieved utilizing Kantorovich-Rubinstein similarity, which provides:

$$W(P_r, P_g) = sup_{f \in \mathcal{F}} E_{x\sim P_r}[f(x)] - E_{\bar{x}\sim P_g}[f(\bar{x})] \qquad (3)$$

where $\mathcal{F}$ is the set of 1-Lipschitz functions. In the process, utilizing W(Pr, Pg) as an allotment of distance training a GAN will adjust the min-max objective function to be expressed as:

$$\min_G \max_{D \in \mathcal{F}} E_{x\sim P_r}[D(x)] - E_{\bar{x}\sim P_g}[D(\bar{x})] \qquad (4)$$

3.1.3 The Convolutional Neural Networks (CNNs)

Convolutional brain networks (CNNs) techniques are changed versions of multi-facet visualizations that are coordinated in a different environment road. The hierarchical example in the data permits them to construct progressively complex examples using lesser, more straightforward example parts. CNN contains of numerous layers, consisting of convolutional, and pooling, with completely joined layers introduced in Figure 10. Convolutional layers include a gathering of learnable pieces that span the whole information. Channels are contorted along the entered data set to result in a 2D activation map by figuring the dab item among its entries against inputs. The pooling operation

measurably reviews the convolutional result. DeepAnt [12][14] CNN-based model It doesn't require broad data in the preparation stage, so it is successful. This model identifies even little deviations in time-series plans and could bargain utilizing low levels of data contamination (under 5%) in an unaided way. The anomaly location model may be employed for univariate and multivariate time series and could recognize abnormalities like point eccentricities, and coherent inconsistencies, with conflicts ,Figure 10 shows the construction of the CNN algorithm model [12].

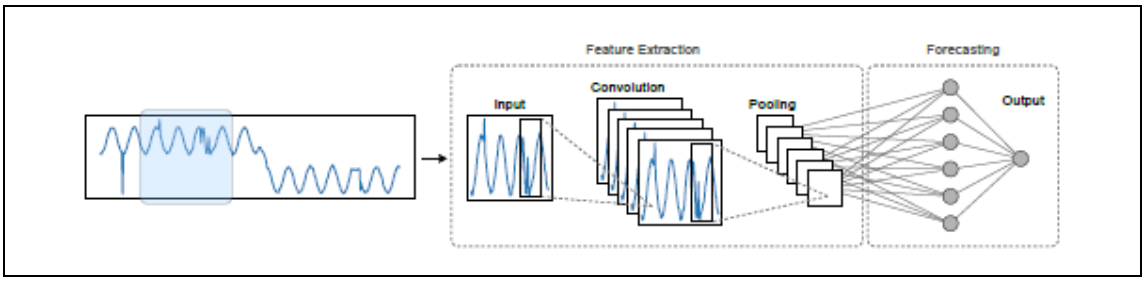


Figure 10: Construction of the (CNN) algorithm model [12].

Regarding Figure 10, we could observe that the CNN technique might also be motivated to produce other types of forecasting deep learning approaches such as the temporal convolutional network (TCN). They are utilized so they might adjust to sequential information. Most CNN-based models employ TCN for time series anomaly detection. Anyhow, it is crucial to note that there is no unique approach or algorithm that could detect all kinds of fake images with 100% accuracy, and as technology advances, so do techniques for creating convincing fake images. Hence, it is necessary to employ a combination of techniques with personal expertise to analyze fake pictures and avoid them growing. The internal details of the CNN algorithm technology consist of a set of several layers or main sections so that each section works to accomplish a specific task about the data entered into this algorithm, which will be briefly explained in the following paragraphs: Convolutional Layer, Pooling layer, activation functions, dropout, loss functions.

3.1.4 Graph Neural Network (GNN).

In the past few years, researchers have suggested extracting spatial information along the multivariate time series (MTS) as well as constructing the graph structure. Then the time series anomaly detection problem is transformed to detect time series anomalies given their graph structures GNNs were utilized to model such graphs. The GNN structure is illustrated in Figure 11.

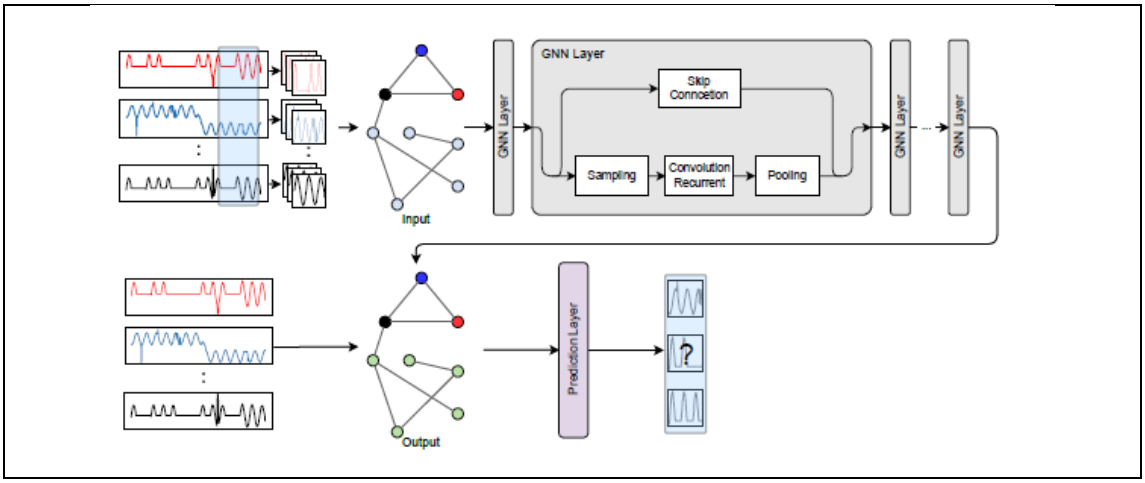


Figure 11: The Graph Neural Network (GNN) essential construction for MTS anomaly recognition which might get familiar with the connections (correlations) among measurements and anticipate the normal way of behaving of time series. [7].

Referring to Figure 11, we might conclude that in the multivariate time series (MTS) anomaly detection models, every dimension (metric) is represented as a single node in the Graph we determine our set of nodes as $V = 1, \ldots, d$. E represents the edges of the graph and indicates correlations that are learned from MTS. For node u/inV, the message passing layer results in the following for iteration $k + 1$:

$$h_u^{k+1} = UPDATE^K\left(h_u^k, m_{N(u)}^k\right), \qquad (8)$$

$$m_{N(u)}^k = AGGREGATE^K(h_i^k, \forall i \in N(u)) \qquad (9)$$

Whereas h^k is the embedding regarding every hub with (u) as the set of neighboring areas of hub u. The capability of GNNs to learn spatial models improves the modeling of multivariate instant series info including Links. Generally, GNNs suggest that the state of every hub is concerned with the states of its neighbors [5]. A broad range of GNN structures have been suggested, implementing various kinds of message passing. Chart A graph convolution network (GCN) [15]. schemes the feature representation of a hub by grouping its neighbors in a unique step. Graph Attention Networks (GATs), rely on such a method, but instead of using a simple weight function for every Neighbor, utilize an attention function to evaluate various weights for every neighbor [5][16].

4. System architecture of the proppsed scheme

After reviewing the most important methods and techniques used to eliminate anomalies in time series data, through prediction processes, we review the programming details and operational specifications for implementing and applying these techniques to applied models of data sets to test the research idea. The hybrid deep learning RNN-LSTM algorithm has been nominated as a model for the proposed model of the technique implemented in this study. Then the flow chart of the suggested anomaly detection of time series data mining in wireless sensor networks using deep learning techniques system model methodology will be illustrated in Figure 12.

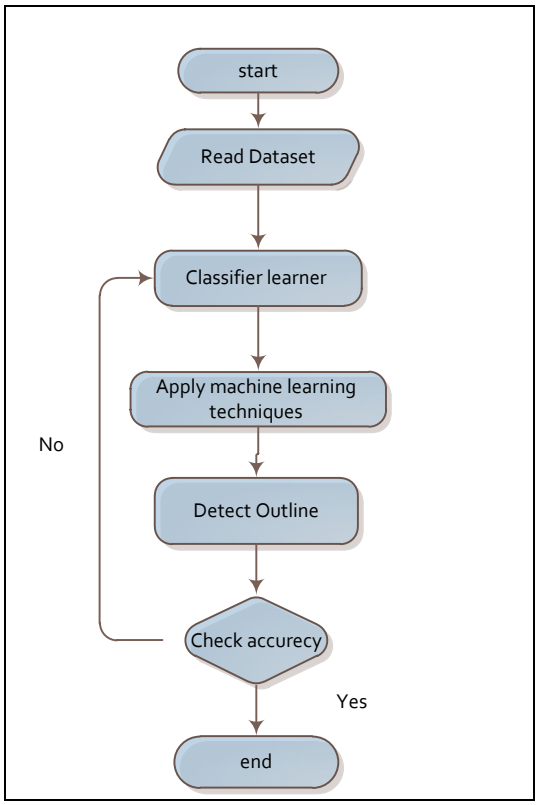


Figure 12: The flow chart of the suggested model methodology.

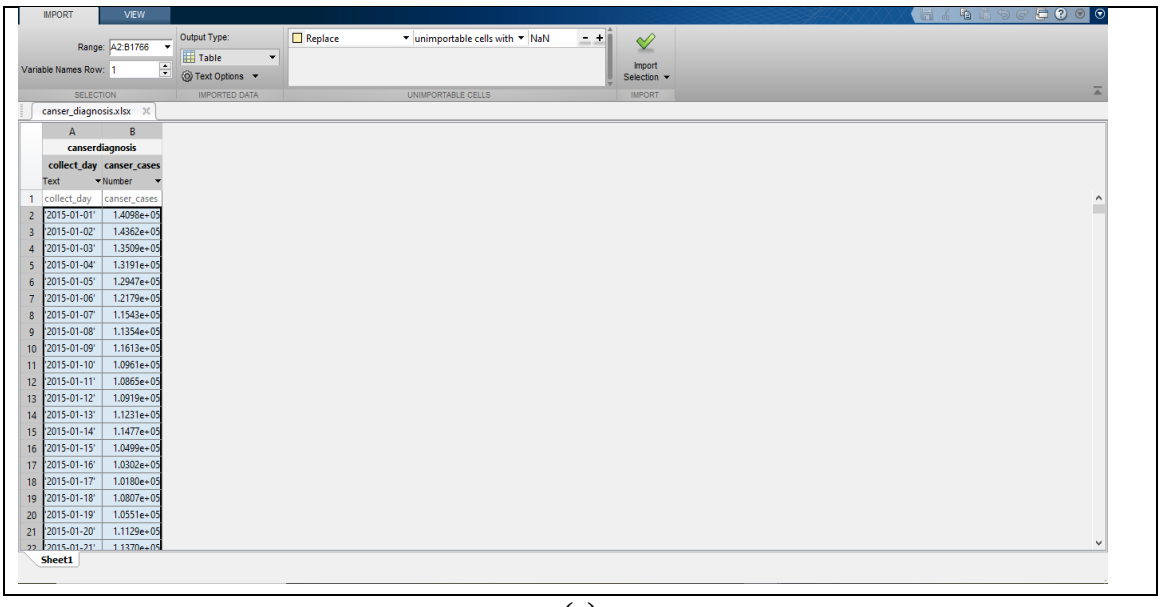
From Figure 12, the proposed model procedure will start using the software by preparing and reading datasets using the Read Dataset option utility. Next, Classifier Learner will be used which will provide the necessary machine learning programming tools. Next, machine learning techniques will be applied using the classifier utility so that the loaded datasets are trained to find the appropriate results. Moreover, the regularization state (data energy) detection option will check for leakage points in the trained datasets, and finally, the leakage detection accuracy will be gained from the classification and training results to run the machine learning algorithms. Finally, it will check the overall steps and finish the processing. Through the implementation of the suggested models for the idea of the research topic, detecting anomalies in time series for data mining and deep learning techniques, the recommended RNNLSTM deep learning technique was implemented on a set of experimental input datasets, which represents readings of cancer cases in a hospital during specific periods with the presence of an anomaly.

4.1 RNN-STM Deep Learning Model

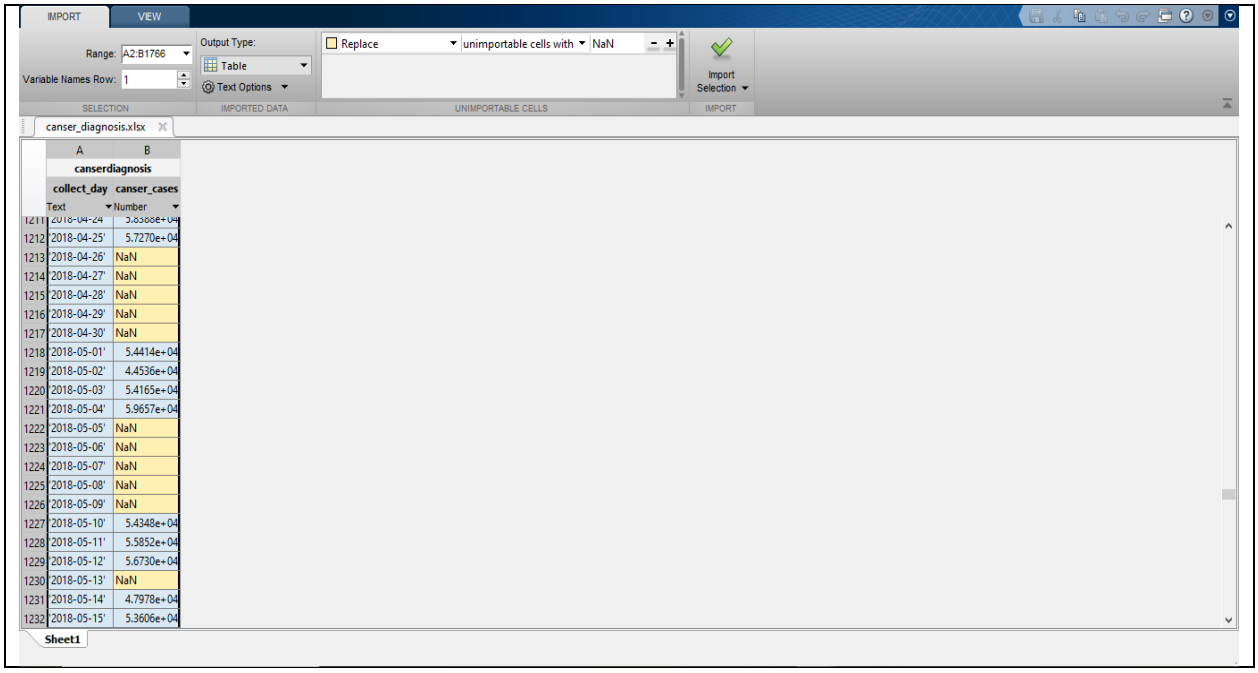
As we previously mentioned, the LSTM algorithm is a type of deep learning RNN technique characterized by its high efficiency and accuracy, in addition to the speed in conducting exploration operations and predicting upcoming values based on the input readings, in addition to its ability to eliminate anomalies. The fundamental thought behind an LSTM network is that it utilizes a model of powers that deals with the info entering and leaving memory modules in the network. Such doors can determine what sections of the sequence to sustain or dispose of, subsequently upgrading the existence of powers in sequence prediction causes. An RNN utilizing LSTMs could be trained in a supervised approach on a set of training sequences, utilizing an optimization algorithm such as gradient descent with backpropagation over the period to evaluate the needed gradients through the optimization operation.

4.2. Model Simulation & Design

In The Dataset Structure of the input readings, Figure 13 shows some sample readings for the input data used in this test.



(a)



(b)

Figure 13: Readings of the input dataset utilized in this examination program code, (a) Cancer reading samples, (b) Anomaly cancer samples.

As we explained, these datasets will indicate the number of people infected with cancer for each period in one of the health institutions. Such data contains readings for years in periods, with incomplete and incorrect readings in some of its contents. We could consider them as cases of anomalies, as these data will represent a time series in which the anomalies are to be corrected and the readings to be predicted for the number of people infected with the same cancer for the coming years. This data will be read through the proposed RNN-LSTM deep learning algorithm, which will represent the set of inputs for the layers of this algorithm to perform the process of eliminating abnormal and incomplete readings and predict future results for the coming years.

5. The achieved results

By implementing the RNN-LSTM deep learning algorithm with the specifications illustrated previously, and with the entered dataset shown in Figure 13, the LSTM algorithm will be trained to process the entered data to find the next incoming predicted reading sequences and cancel the anomaly presented data samples. Figure 14 shows the training progress of the RNN LSTM deep learning algorithm with the entered cancer datasets.

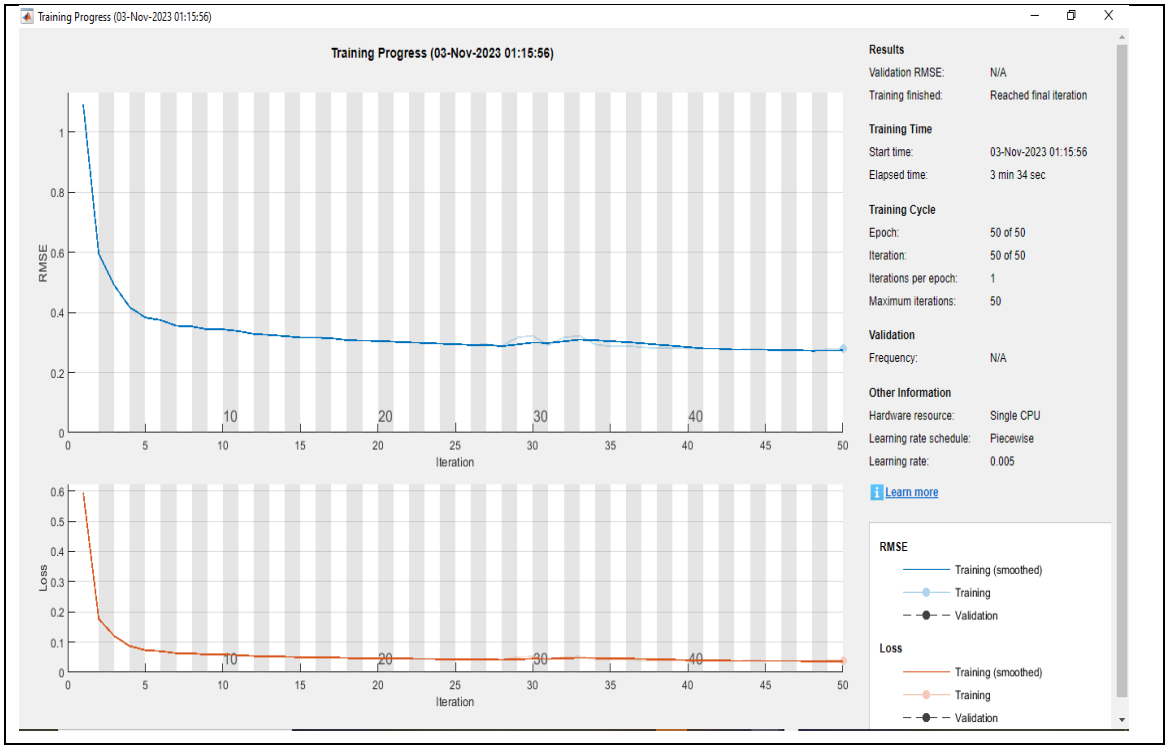


Figure 14: The training progress of the RNN LSTM deep learning algorithm with the entered cancer datasets.

By looking at Figure 14, we can observe the training progress of the proposed algorithm for test carving data inputs, where we notice the success of the algorithm in conducting the data training process predicting the future values of cancer patients' cases and ignoring cases of abnormal readings. The root mean square error (RMSE) readings show values as low as 0.1, while the data loss readings show a very small percentage of up to 0.05, which indicates the accuracy of the training and the success of the prediction process with a high efficiency of up to 99.95%. Also, the results of the forced data produced by applying the RNN-LSTM deep learning technique have been obtained and compared with the observed readings from the trained cancer datasets as displayed in Figure 15.

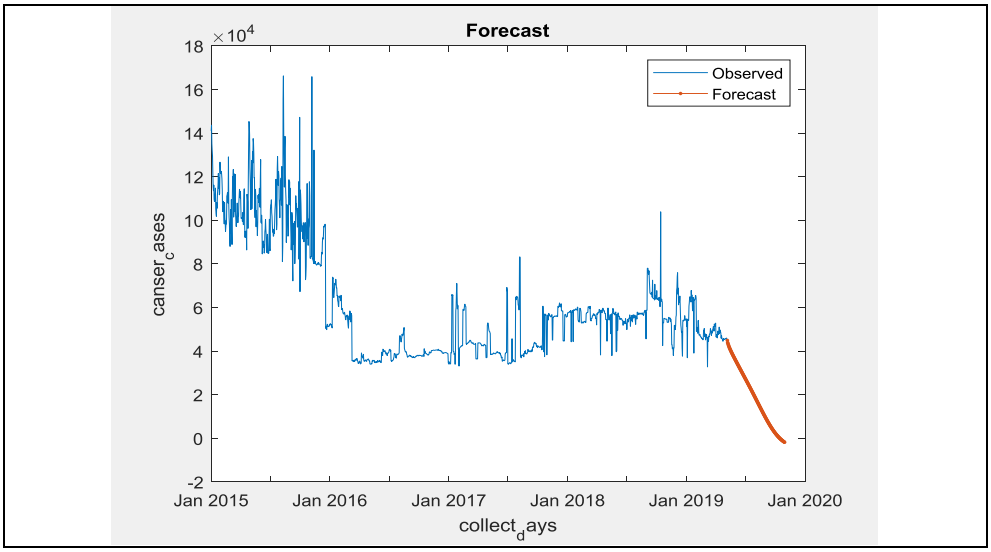


Figure 15: The forced data produced by applying the RNN-LSTM deep learning technique have been obtained and compared with the observed readings from the trained cancer datasets.

Referring to Figure 15, we can notice the prediction results for the outputs of the proposed deep learning algorithm. We recognize the correspondence between the results of the forecast chart shown in red color and the forecast readings chart shown in blue color. The results of the predictive forecasts, shown in red color, appear free of the distortions (anomalies) that were clearly visible in the results of the observed readings chart, shown in blue color, as presented in the figure above. After that, the training results are recorded for a small set of readings for the first 180 readings, and the resulting readings of the prediction output are compared with the original monitored readings, as shown in Figure 16.

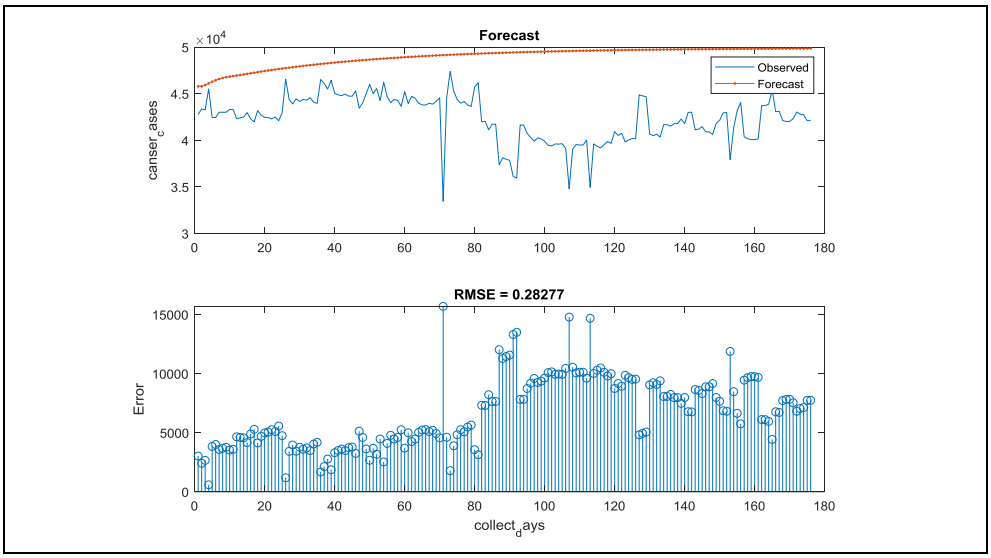


Figure 16: The recorded training results for a small set of readings for the first 180 datasets, with the resulting outcomes of the prediction output compared with the original observed readings.

By reviewing the training results of a small set of readings for the first 180 readings, the resulting prediction output readings were compared with the original observed results. The predictive results for cancer cases appear in red, clearly free of distortions and anomalies, compared to the originally observed readings presented in blue color in Figure 16. Also, the obtained RMSE of the trained data has been recorded to be 0.282777 utilizing the deep learning RNN-LSTM algorithm which indicates a perfect prediction and anomaly detection for the entered examined dataset.

6. Results discussions

At the end of this work, and through the details explained on the subject of detecting anomalies in time series using deep learning techniques for data mining, and after implementing the proposed RNN-LSTM deep learning technique on the data model for cancer statistics, the results obtained were studied in terms of efficiency and accuracy of performance. The training progress of the suggested algorithm was observed for testing with the implemented data inputs, as we note the success of the algorithm in conducting the training process on the data, predicting the future values of cancer patient cases, and ignoring cases of abnormal readings. The Root Mean Square Error (RMSE) readings show values as low as 0.1, while the data loss readings show a very small percentage of up to 0.05, which indicates the training accuracy and success of the prediction process with a high efficiency of up to 99.95%. Also, by reviewing the training results of a small set of readings for the first 180 readings, the resulting prediction output readings were compared with the original observed results of training the implemented algorithm. The predictive results for cancer cases were clearly free of distortions and anomalies, compared to the original observed readings. The RMSE value obtained from the trained data was recorded to be 0.282777 using the proposed deep analysis technique. The learning results of the RNN-LSTM algorithm showed high prediction

readings and anomaly detection for the examined data set, which was entered with high efficiency and accuracy.

7. Conclusions

In this analytical study, the various techniques and methods used in the subject of detecting anomalies in time series using deep learning techniques for data mining were reviewed, and through the details described in this research. The most important methods and strategies for analyzing and classifying data were reviewed, and the types of data used in the wireless sensor network (WSN) environment were identified, as methods for mining them. In addition, various anomaly problems in various data series were studied, along with identifying most of the techniques and strategies used in big data mining, such as machine learning techniques deep learning algorithms, and others. The proposed deep learning RNN-LSTM technique has been nominated for implementation on a data model for cancer statistics to predict and detect potential anomalies and errors in this data. The results obtained were studied in terms of efficiency and accuracy of performance, obtaining a high-efficiency rate for prediction and detection of anomalies that reached 99.95%, with the RMSE value obtained from the trained data recorded at 0.282777.

References

1. A. Garg, W. Zhang, J. Samaran, R. Savitha, and C. S. Foo, "An Evaluation of Anomaly Detection and Diagnosis in Multivariate Time Series," *IEEE Trans Neural Netw Learn Syst*, vol. 33, no. 6, pp. 2508–2517, Jun. 2022, doi: 10.1109/TNNLS.2021.3105827.
2. D. Park, Y. Hoshi, and C. C. Kemp, "A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder," *IEEE Robot Autom Lett*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018, doi: 10.1109/LRA.2018.2801475.
3. C. Zhang *et al.*, "A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 1409–1416, Jul. 2019, doi: 10.1609/AAAI.V33I01.33011409.
4. A. Deng and B. Hooi, "Graph Neural Network-Based Anomaly Detection in Multivariate Time Series," *AAAI Conference on Artificial Intelligence*, vol. 5A, pp. 4027–4035, 2021, doi: 10.1609/AAAI.V35I5.16523.
5. X. Xie, B. Wang, T. Wan, and W. Tang, "Multivariate Abnormal Detection for Industrial Control Systems Using 1D CNN and GRU," *IEEE Access*, vol. 8, pp. 88348–88359, 2020, doi: 10.1109/ACCESS.2020.2993335.
6. A. Deng and B. Hooi, "Graph Neural Network-Based Anomaly Detection in Multivariate Time Series," *AAAI Conference on Artificial Intelligence*, vol. 5A, pp. 4027–4035, 2021, doi: 10.1609/AAAI.V35I5.16523.
7. M. Hu *et al.*, "Detecting Anomalies in Time Series Data via a Meta-Feature Based Approach," *IEEE Access*, vol. 6, pp. 27760–27776, May 2018, doi: 10.1109/ACCESS.2018.2840086.
8. X. Chen *et al.*, "DAEMON: Unsupervised anomaly detection and interpretation for multivariate time series," *Proc Int Conf Data Eng*, vol. 2021-April, pp. 2225–2230, Apr. 2021, doi: 10.1109/ICDE51399.2021.00228..
9. B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Comput*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001, doi: 10.1162/089976601750264965.
10. P. Comon, "Independent component analysis, A new concept" *Signal Processing*, vol. 36, no. 3, pp. 287–314, Apr. 1994, doi: 10.1016/0165-1684(94)90029-9.

11. K. Choi, J. Yi, C. Park, and S. Yoon, “Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines,” *IEEE Access*, vol. 9, pp. 120043–120065, 2021, doi: 10.1109/ACCESS.2021.3107975.

12. S. Chauhan and L. Vig, “Anomaly detection in ECG time signals via deep long short-term memory networks,” *Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015*, Dec. 2015, doi: 10.1109/DSAA.2015.7344872.

13. J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, “USAD: UnSupervised Anomaly Detection on Multivariate Time Series,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 20, pp. 3395–3404, Aug. 2020, doi: 10.1145/3394486.3403392.

14. G. Pang, C. Shen, L. Cao, and A. Van Den Hengel, “Deep Learning for Anomaly Detection,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, Mar. 2021, doi: 10.1145/3439950.

15. Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng, “Learning Graph Structures With Transformer for Multivariate Time-Series Anomaly Detection in IoT,” *IEEE Internet Things J*, vol. 9, no. 12, pp. 9179–9189, Jun. 2022, doi: 10.1109/JIOT.2021.3100509.

16. M. Hu, X. Feng, Z. Ji, K. Yan, and S. Zhou, “A novel computational approach for discord search with local recurrence rates in multivariate time series,” *Inf Sci (N Y)*, vol. 477, pp. 220–233, Mar. 2019, doi: 10.1016/J.INS.2018.10.047.