Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

Extended Representation Learning Based Neural Network Model for Outlier Detection

Sidratul Muntaha[1], Sohana Jahan[2], Md. Anwarul Islam Bhuiyan[3*]

[1,2,3]Department of Mathematics, University of Dhaka, Bangladesh India.*

Email: [1]sidratulmuntaha.du.bd@gmail.com, [2]anwarulislam1810@gmail.com
Corresponding Email: [3]sjahan.mat@du.ac.bd*

Received: 02 June 2024 **Accepted:** 16 August 2024 **Published:** 01 October 2024

Abstract: ***Outlier detection problems have drawn much attention in recent times for their variety of applications. An outlier is a data point that is different from the rest of the data and can be detected based on some measure. In recent years, Artificial Neural Networks (ANN) have been used extensively for finding outliers more efficiently. This method is highly competitive with other methods currently in use such as similarity searches, density-based approaches, clustering, distance-based approaches, linear methods, etc. In this paper, we have proposed an extended representation learning based neural network. This model follows a symmetric structure like an autoencoder where the dimensions of the data are initially increased from their original dimensions and then reduced. Root mean square error is used to compute the outlier score. Reconstructed error is calculated and analyzed to detect the possible outliers. The experimental findings are documented by applying it to two distinct datasets. The performance of the proposed model is compared to several state-of-art approaches such as Rand Net, Hawkins, LOF, HiCS, and Spectral. Numerical results show that the proposed method outperforms all of these methods in terms of 5 validation scores, Accuracy (AC), Precision (P), Recall, F1 Score, AUC score.***

Keywords: Auto Encoder Model, Machine Learning, Neural Networks, Outlier Detection.

1. INTRODUCTION

In recent times, outlier detection or anomaly detection [32] [1] [23] problems have drawn much attention for their variety of applications [3]. An outlier is a data point that differs significantly from the rest of the data. It deviates so much from other observations that it arouses suspicions as if it may have been generated by a different mechanism [2]. Outliers have a major impact on both statistical analysis and machine learning models.

Although outliers are often considered as errors or noise, they may carry important information. So, outliers should be treated carefully. A data point may have many features and to detect an

Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

outlier effectively, finding the appropriate combination of features that will enable the algorithm to produce the greatest results in terms of accuracy and computational efficiency is a significant issue. Therefore, an obvious answer to this issue is provided by neural network learning methods. Machine learning algorithms [20] specially Neural network models have emerged as a powerful tool for outlier detection, as they can learn to encode and decode complex data distributions and identify anomalies that deviate significantly from the learned distribution. We present our two-part extended representation learning based neural network, the encoder network and the decoder network. An encoder network converts the input data to a lower-dimensional representation and a decoder network converts the lower-dimensional representation back to the original input space for outlier detection. To find outliers in the data, the difference between the input and the reconstructed output is utilized as a measure of the reconstruction error. We designed our extended learning neural network model, which encodes and decodes the data and computes the reconstruction error. The architecture of the neural network model should be chosen based on the specific problem and data at hand and should be tuned carefully to avoid overfitting and ensure optimal performance. The advantage of using a neural network model is that it can often improve the accuracy and robustness of the predictions, capture different aspects of the data, and reduce the risk of overfitting. However, a neural network model could cost more to compute and might need additional hyper parameter tuning.

2. RELATED WORK

In the community of data mining, there has been extensive research on the issue of outlier analysis. Many real life situations can be formulated and solved as anomaly detection problem [7], [9], [10]. If we train a machine learning algorithm for other data mining tasks such as clustering or classification where the data contains outliers, then the model can be distracted by the outliers which may lead to higher classification error. So, analyzing data and looking for outliers beforehand is very important for such data mining techniques as well. Reducing dimension of data using dimension reduction techniques [26] [27] [36] also may be helpful for better performance. In recent years, a variety of techniques for outlier detection have been presented in [4][5][6] [12][14][18][19][28][30][33] . Several techniques, such as the distance-based method [11, 34], the density-based method [22], the linear method, and spectral methods, have been proposed by the researchers. Using feed- forward multilayer neural networks, replicator neural networks (RNNs) have been proposed in [15] [24]. The Boosting-Based Autoencoder Ensemble technique (BAE) is another option [16]. Additionally, a Rand Net neural network model has been suggested by the author in [17]. For anomaly identification, there is also a robust deep autoencoder [8]. The Rand Net model [17] suggests an autoencoder ensemble model that incorporates an adaptive sampling technique. Hawkins [24] searches vast multivariate collections for outliers using a replicator neural network (RNN). They essentially employ a feed-forward multilayer perceptron. LOF [19] stands for Local Outlier Factor. They put forward a strategy for identifying outliers in a multidimensional dataset. In essence, each object's confined neighborhood is connected to density-based clusters. HiCS (High Contrast Subspaces) [13] suggests a first measure for the contrast of subspaces, a unique subspace search technique that chooses high contrast subspaces for density-based outlier rating. Spectral [29]

Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

suggests the LODES approach, which finds outliers by combining local density-based methods and spectral techniques. Finding the best features of the data is a crucial issue for any outlier identification technique. In this work, we have proposed a neural network-based model that finds the ideal set of features that will enable the algorithm to produce the greatest results in terms of accuracy and computing efficiency.

Motivation of this Research

While some neural network models are effective in reducing the dimension of the data, there is indeed a trade-off, and some information loss can occur. The encoder's task is to capture the most important features or patterns in the input data and represent them in a reduced-dimensional space. Nevertheless, there's always a chance that certain details and features from the original material will be lost due to compression. We have conducted experiments using various neural network model architectures. Analyzing the results of these methods, we have observed that, if we initially reduce the dimension of the actual data set, there is a chance to lose some information here. Taking this into consideration we have initiated to develop a model that expands feature space by initially increasing the dimension of the data set. This approach captures additional information before the final compression for dimensionality reduction occurs in the encoder section to train the model. After that, we symmetrically generate the decoder section. Our goal is to capture more information or create a richer feature space.

Contribution of this Research

The contribution of this work can be summarized as follows:

1. We have proposed an extended representation learning based neural network (ERLNN) model.
2. Outlier score is calculated to detect the outliers. In calculating the outlier score, Root Mean Square Error (RMSE) is proposed instead of using mean square error as it helps to find the large outlier score which corresponds to outliers or anomalies present in the data
3. By examining the training data and performing a more effective error analysis, we have addressed a dynamic threshold for identifying the outliers.
4. The proposed model is applied to several datasets collected from UCI repository. In this paper, we have documented the experimental findings of two datasets.
5. The performance of the proposed model is compared to several state-of-art approaches such as Rand Net, Hawkins, LOF, HiCS, and Spectral.
6. The results are summarized in terms of 5 validation scores, Accuracy (AC), Precision (P), Recall, F1 Score, AUC score.

Paper Organization

The rest of the article is organized as follows.

In the next section, we have introduced our proposed model. Short description of the data which are collected from the (UCI) repository [31] and the data preprocessing s is provided in the next section. In section 5, experimental results and comparisons between the algorithms are demonstrated and discussed. Finally, the paper is concluded in Section 5.

Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

3. MATERIALS AND METHODS

Autoencoder models can be used to detect outliers in a dataset by first training the autoencoder on a set of normal, non-outlier data points, and then using the trained model in order to recreate the test data. The reconstruction error, or the difference between the input data and its reconstructed output, can then be used as a measurement of how accurately the autoencoder can depict the input data. Data points that have a high reconstruction error are likely to be outliers since they do not fit well with the learned representation of the normal data as shown in Fig. 1. Thus, a threshold can be set on the reconstruction error, and data points with a reconstruction error above this threshold can be flagged as potential outliers. The methodology of a model for outlier detection is given in Fig. 1.

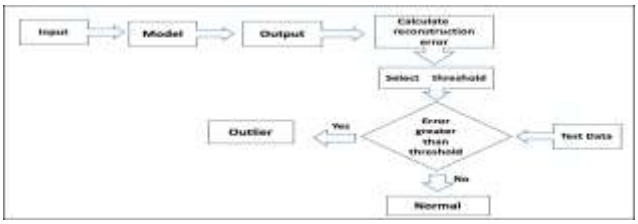


Figure 1. Methodology Flowchart

2.1 Proposed Model

Model Architecture

We represent an auto encoder-type model in which we initially increased the dimension and then reduced it. It is noted here that our input layer contains exactly the same number of nodes as the dimension of the data set. As our goal is to reconstruct the data, our output layer has the same number of nodes. The basic structure of our model is shown in Fig. 2

Mathematical Formulation

Suppose a dataset with m data points, where $X = \{X_i\}_{i=1}^{m}$the input data is. Let $Y = \{Y_i\}_{i=1}^{m}$ be the reconstructed data or output data, $n$is the number of features that means each$X \in \mathbb{R}^n$.
Then, for the forward pass, we use the equation $Z = f\big(E(X)\big)$, where E is the encoder function, f is the activation function, and Z is the latent representation. The data is reconstructed by the decoder: $Y = g\big(D(Z)\big)$, where D is the decoder function, g is the activation function, and Z is the latent representation.

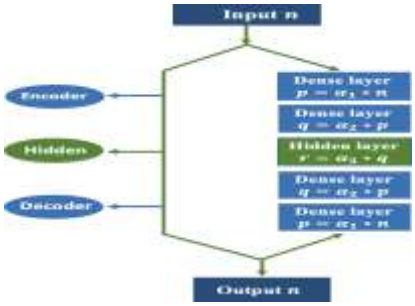


Figure 2. Structure of the proposed Model (here $p > n, p > q > r$ and $r < n$)

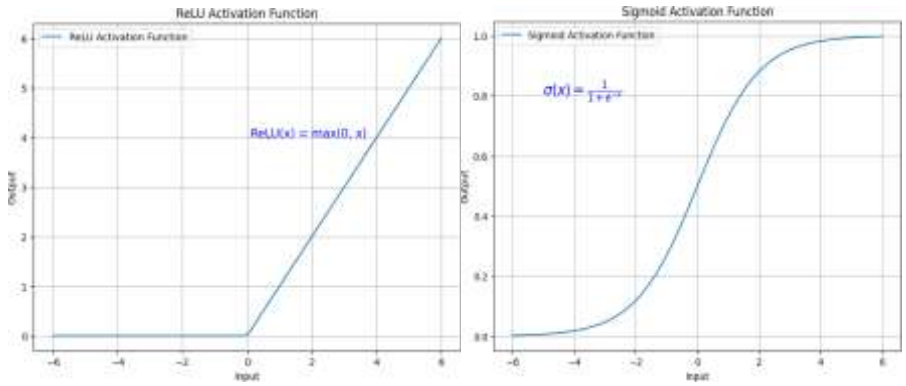
Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

For reconstruction error, we use the MSE function

$$e = L(X, Y) = \frac{1}{m} * \sum_{i=1}^{m} \left|\left|X_i - Y_i\right|\right|^2 = \frac{1}{m} * \sum_{i=1}^{m} {e_i}^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Where e_i is the reconstruction error relative to X_i, m is the number of data points in the data set. Backpropagation is used in the backward pass to calculate the gradients of the loss function with respect to the model parameters. For developing our model, we use Adam optimizer as Adam uses adaptive learning rates for each parameter, adjusting them during training. This adaptability allows it to handle different features and gradients individually. We compute the mean square error (mse) loss defined in Eq. (1) for the loss function calculation, and we ensure that our loss decreases at each epoch. For optimization, backpropagation allows the network to adjust its internal parameters (weights and biases) based on the difference between its predictions and the actual outcomes in the training data.

For the activation functions, we choose the Rectified Linear (ReLu) units $R(x) = \max(0, x)$ (Fig. 3(a)) and the Sigmoid function $\delta(x) = \frac{1}{1+e^{-x}}$ (Fig. 3(b).



(a) ReLU Activation Function (b) Sigmoid Activation Function

Figure 3. Activation Function

We have used different activation functions in different layers to balance their advantages and disadvantages. It is observed in related studies and practice that the combination of activation functions gives better performance than a fixed choice. To avoid the vanishing gradient problem caused by the sigmoid function and its expensive computational complexity, we have used ReLU in most of the layers. However, recent research says, the ReLU unit suffers from a dying ReLU problem which causes the inactivation of some neurons at some point of the training process. This happens because during training sometimes a large gradient trigger a weight update through a ReLU neuron. As a result, the network may continue to give the same output over iterations. To overcome this situation, we have used sigmoid function at the end of the network. We use the sigmoid function at the end of the network and the ReLU activation function in the middle of the layers in the network [17]. The structure of the proposed network is given in Fig. 4.

Our ERLNN model is trained only with normal data sets in order to identify patterns in the data set. An outlier score is determined for each data point by taking the root mean square error of the original data and the predicted data as the reconstructed error. Using the trained model, the

Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

testing data set, which contains the outlier, is reconstructed and predicted. The outlier score is calculated as follows:

$$\text{Outlierscore, } O_i = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(X_{ij} - Y_{ij}\right)^2} \quad \text{.....(2)}$$

Here:

n is the number of features in a data point.
$X_i \in \mathbb{R}^n$ is the actual or true value for the i-th data point.
$Y_i \in \mathbb{R}^n$ is the predicted value for the i-th data point

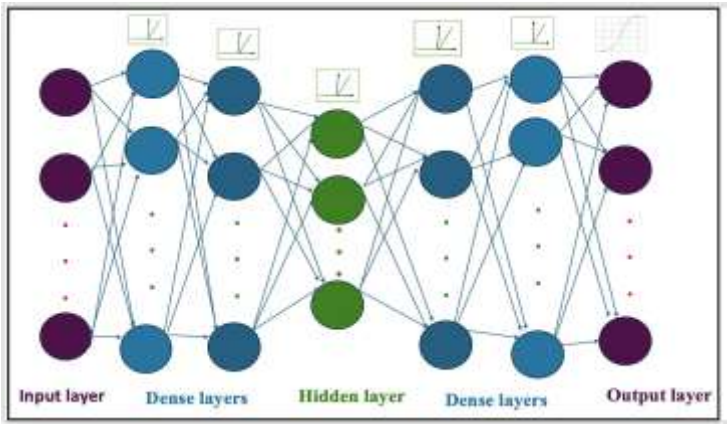


Figure 4. Structure of proposed network

Effective Threshold Selection

The outlier threshold is chosen during the model's training process using the outlier score of the training data obtained by Eq. (2). We calculate the median and the Inter-Quartile Range (IQR) from the outlier scores and use the following formula to determine the threshold.

$$\text{Threshold} = \text{Median} + 1.5 * \text{IQR} \quad \text{..... (3)}$$

$$IQR = Q_3 - Q_1 \quad \text{.....(4)}$$

Q_1 is the outlier score of data point at the position $\frac{m+1}{4}$ and Q_3 is the that of at the position $\frac{3(m+1)}{4}$ where m is the number of data points in the dataset. For testing data, if the outlier score is higher than the threshold then it is treated as an outlier.

Numerical Experiment

Dataset Description

It is common practice to evaluate the feasibility of outlier detection methods using classification-oriented data. In this paper, we use two data sets: one is the E. coli data set, and the other is the cardiotocography (CTG) data set collected from the UCI Machine Learning Repository [31]. These datasets are designed for the classification problem, E. coli dataset contains 8 classes and they are cp, im, imS, imU, imL, Om, omL and pp. For our work, we used classes omL, imL and ImS as outliers and the rest were included as normals. On the other hand, the CTG data set can

Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

be categorized according to morphologic pattern (A, B, C, etc.) or fetal condition (N=normal, S=suspicious, P=pathologic). In this work, the suspicious group was eliminated. The pathologic class was designated as the outliers, and the normal class as the inliers.

Data Set Preprocessing

Data preprocessing is a prerequisite for conducting the tests. So, before we go to examine the results by fitting training data to the proposed model, we split the data set as follows:

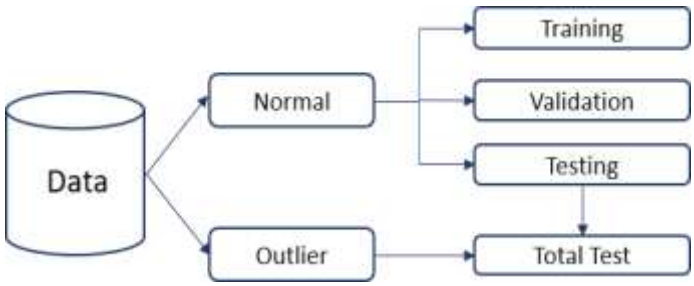


Figure 5. Data set split for Outlier Detection

Data Split	Number of data points
Normal data	326
Training data	260
Data for validation	33
Testing data	33
Outlier	9
Total Testing data	42

Table 1. Train-validation-test Split of the Ecoli Data Set

Data Split	Number of data points
Normal Data	1655
Training data	1241
Data points for Validation	207
Testing data	207
Outlier	176
Total Testing data	383

Table 2. Train-validation-test Split of the CTG data set

We split the normal E.coli data and take 80% for training and from the rest of 20% we take 50% for validation and the rest of the 50% for testing. Then we add 9 outliers to our testing data and generate our total testing data set. In the CTG data, we take 75% of the normal data for training, and from the rest of 25% we take 50% for validation and the rest of the 50% for testing. Then we add 176 outliers to our testing data and generate our total testing data set.

Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

Structure of Hidden Layers

This section contains a list of the experiments that were conducted on the E. coli dataset and the CTG dataset using the proposed methodology presented in Fig. 1, the network Fig. 4 and the model structure Fig. 2. For the experiment, we chose $\alpha_1 = 30$, $\alpha_2 = 1/3$ and $\alpha_3 = 1/14$. This means first we have increased the dimension of the data 30 times its original dimension and then it was decreased until the hidden layer. The data is reconstructed in the output layer by following a symmetric approach on the two sides of the hidden layer. For outlier detection, the threshold value is determined by Eq. (3-4).

Evaluation Metrics

In our data sets, we labeled our data as normal data and outlier data. To evaluate our model's performances, we generate true positives (TP), where the model correctly predicts the positive class. Similarly, we generate true negative (TN), where the model correctly predicted the negative class, false positive (FP), where the model incorrectly predicted the positive class when the actual class was negative, and false negative (FN), where the model incorrectly predicted the negative class when the actual class was positive. These results for our data sets have been shown through a confusion matrix. A confusion matrix is a matrix that summarizes the performance of a machine learning model on a set of test data, which aims to predict a categorical label for each input instance. The confusion matrices for each data set are given in the next section along with other numerical measures.

To determine the accuracy of the proposed model five validation indices (VI) accuracy (AC), precision (PR), recall (RE), F1-measure and AUC score. An algorithm having higher values of AC, PR, RE, F1 and AUC implies having better performance than others. The VI indices AC, PR, RE, F1 and AUC are defined as follows by:

i. Accuracy=(TP+TN)/(TP+TN+FP+FN)
ii. Precision = TP/ (TP + FP)
iii. Recall = TP/ (TP + FN)
iv. F1 = 2(Precision*Recall)/ (Precision + Recall).
v. The AUC score ranges in value from 0 to 1, where 0 represents the model prediction is 100% wrong. On the other hand, if the model has AUC score of 1 then it represents 100% correct prediction.

4. RESULT AND DISCUSSION

Results from the E. Coli Data

The model used for Ecoli data is given in Fig. 6. The figure shows that we have used two dense layers of sizes 210 and 70 in the encoder part one dense layer of size 5 as a hidden layer and two dense layers in the decoder part of sizes same as in the encoder. After training the model and performing the validation, we have observed that the model decreases the loss function and increases the accuracy. For every epoch, the model's performance is documented.

Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

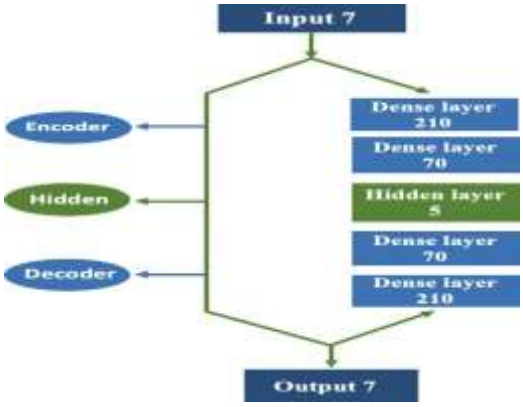


Figure 6. Model for E. coli

Figure 7. MSE loss for E. coli at each epoch

Graphical representation in Fig. 7 shows that the MSE loss is decreasing for each epoch of the training and validation data which indicated the model accuracy as well.

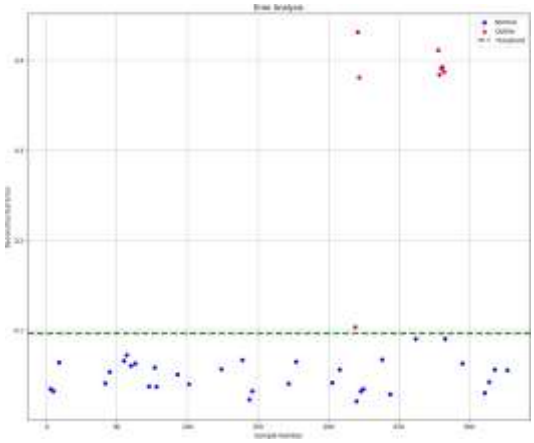


Figure 8. Reconstruction Error of E coli test data

Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

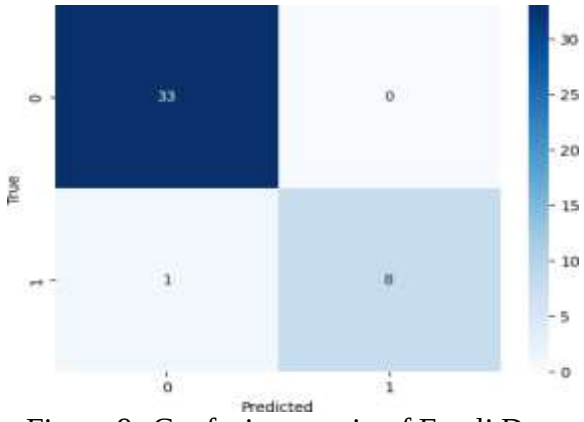


Figure 9. Confusion matrix of E coli Data

The threshold value for this data determined by Eq. (3-4) is 0.025. From the confusion matrix of the E coli data presented in Fig. (9), it can be seen that the model predicts 33 normal data among 33 normal data correctly which is our true positive (TP). The model predicts 8 out of 9 outliers correctly which can be seen from Fig. 8 as well. This number of correctly predicted outliers represents the true negative. The model counts one outlier data as normal data which is a false positive. On the other hand, it doesn't count any normal as an outlier which is our false negative. That means the recognition rate of normal data is 100%, and the recognition of outliers is 88.89%. The Accuracy, Precision, Recall, F1-Score and AUC scores for Ecoli data calculated from the confusion matrix are documented in Table 3.

Term	Model's Result
Accuracy	97.62
Precision	100.00
Recall	88.89
F1 Score	94.12
AUC Score	94.44

Table 3. Performance of Ecoli Data

Term	Model's Result
Accuracy	93.21
Precision	93.60
Recall	91.48
F1 Score	92.53
AUC Score	93.08

Table 4. Performance of CTG Data

Results from the CTG Data

Using the same structure as in the case of Ecoli, the model for CTG data is given in Fig. 10. Here at the encoder part, we used 2 dense layers of sizes 630 and 210 respectively. One hidden layer of size 14 is chosen in the middle. As mentioned earlier, ReLU activation function is used

Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

in each of these layers and sigmoid is used in the output layer.

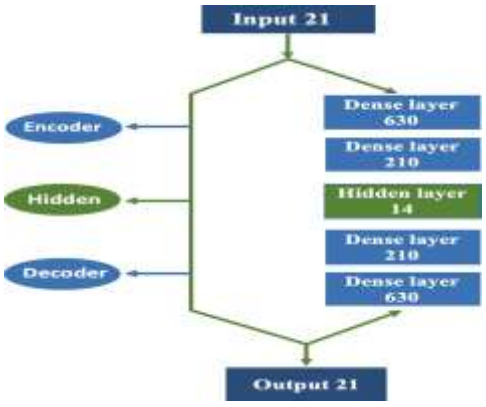


Figure 10. Model for CTG

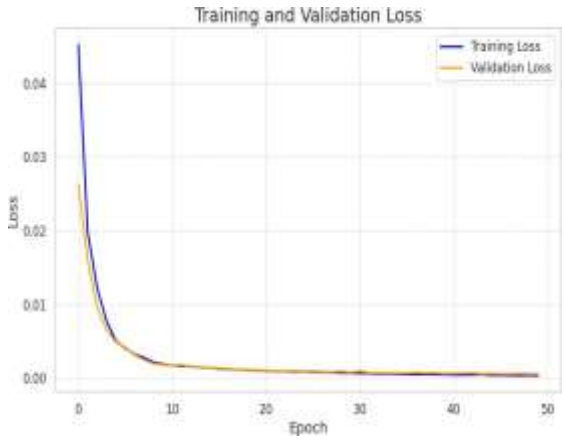


Figure 11. MSE loss for CTG at each epoch

Performing the training and validation on CTG data, we observed that the model decreases the loss function for this data as well which is also visible in Fig. 11. The reconstruction error of the test set of CTG data is shown in Fig. 12. The figure shows the outliers obtained by comparing the outlier score of the data points with the threshold value.

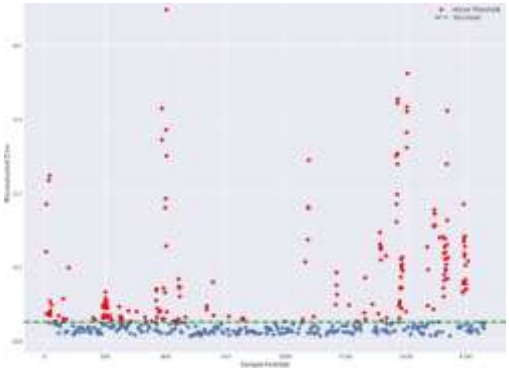


Figure 12. CTG test data reconstruction erro

Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

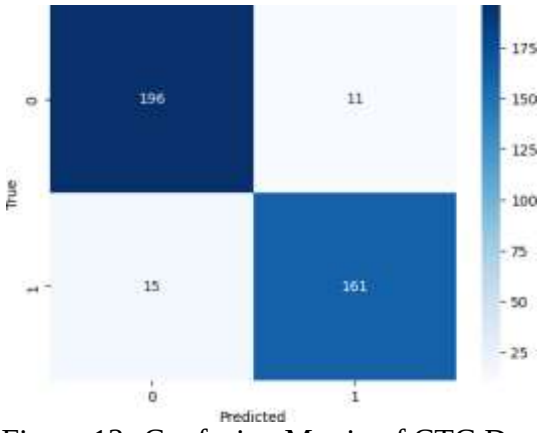


Figure 13. Confusion Matrix of CTG Data

However, the model predicted 15 outliers as normal data which represents false positives and it predicted 11 normal data as outliers which is a false negative. These results are summarized using the four validation indices in table 4.
The performance of the model is compared with that of other state-of-the-art approaches in terms of AUC score which is given in the table 5.

Method	Ecoli data	CTG data
ERLNN model	94.44	93.08
RandNet[17]	85.42	92.87
Hawkins[24]	82.85	92.36
LOF[19]	39.35	50.63
HiCS[13]	53.89	92.37
Spectral[29]	91.81	78.90

Table 5. Comparison of AUC scores obtained by different methods

Table 5 shows that ERLNN model outperforms each of these methods in terms of AUC score. For CTG data Rand Net, performs better than rest of the methods whereas Spectral shows a similar performance for Ecoli data. However, though HicS performs poorly for Ecoli data, for CTG data it works very well.

5. CONCLUSION

An extended representation learning-based neural network (ERLNN) model is proposed. Depending on the datasets that we have used in this work, different architectures of ERLNN models are presented. Initially, the dimension was increased 30 times its original and then the dimension was reduced in the following layers by a ratio $\frac{1}{30}$ and then $\frac{1}{14}$ in the hidden layer.
To detect the outliers, reconstruction error is calculated using root mean square error from the training data and a threshold value is selected using the median and the Inter-Quartile Range (IQR) of the outlier scores. Comparing the outlier scores of each of the testing data against a threshold value, outliers are detected. Numerical experiments were carried out over several

Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

datasets collected from UCI repository. In this article, the results of 2 datasets Ecoli and CTG data are included. The performance is observed using 5 measures, AC, PR, RE, F1, and AUC score. We conclude our work by comparing these two models' performances with several state-of-the-art approaches RandNet, Hawkin's, LOF, HiCS, and Spectral. Numerical results show that ERLNN model obtains the best accuracy compared to related studies. Applying the proposed approach to different datasets, we observed that the model can be improved by adding a structure parameter to efficiently identify the important features. Also error-smoothing methods like the Exponentially Weighted Moving Average Method (EWMA), and Regularization techniques can be explored for further analysis.

6. REFERENCES

1. A. Abhaya, and B. K. Patra, "An efficient method for autoencoder based outlier detection", Expert Systems with Applications, 213(2023), Part A, 2023, pp. 118904.
2. A. Zimek, R. J. G. B. Campello, and J. Sander, "Ensembles for unsupervised outlier detection: challenges and research questions a position paper", Association for Computing Machinery vol.15 (1), pp. 11–22, 2014.
3. B. Dastjerdy, A. Saeidi and S. Heidarzadeh, "Review of Applicable Outlier Detection Methods to Treat Geotechnical Data", Geotechnics, vol. 3(2), pp. 375-396, 2023.
4. C. C. Aggarwal, "An introduction to Outlier analysis", Springer, Cham, pp. 1-34, 2017.
5. C. C. Aggarwal, and S. Y. Philip, "Outlier detection for high dimensional data", Proceedings of the 2001 ACM SIGMOD international conference on Management of data. 2001.
6. C. Mattos, C. Lincoln, G. A. Barreto, and G. Acuna, "Randomized Neural Networks for Recursive System Identification in the Presence of Outliers: A Performance Comparison," Advances in Computational Intelligence: 14th International Work Conference on Artificial Neural Networks, IWANN 2017, Springer International Publishing, 2017.
7. C. Wang, B. Wang, H. Liu, and H. Qu, "Anomaly detection for industrial control system based on autoencoder neural network", Wireless Communications and Mobile Computing, pp. 1–10, 2020.
8. C. Zhou, and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders", Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017.
9. D. M. Rocke, and D. L. Woodruff, "Identification of outliers in multivariate data", Journal of the American Statistical Association vol. 91(435), pp. 1047–1061, 1996.
10. D. Shalyga, P. Filonov, and A. Lavrentyev, "Anomaly detection for water treatment system based on neural network with automatic architecture optimization", arXiv preprint arXiv: 1807.07282, 2018.
11. E. M. Knorr, and T. N. Raymond, "Algorithms for mining distance-based outliers in large datasets", Proceedings of the international conference on very large data bases, 1998.
12. E. M. Knorr, T. N. Raymond, and V. Tucakov, "Distance-based outliers: algorithms and applications", The VLDB Journal vol. 8(3), pp. 237–253, 2000.
13. F. Keller, E. Muller, and K. Bohm, "HiCS: High contrast subspaces for density-based outlier ranking", IEEE 28th international conference on data engineering. IEEE, 2012.

Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

14. G. Dudek, "Autoencoder based Randomized Learning of Feedforward Neural Networks for Regression", 2021 International Joint Conference on Neural Networks (IJCNN). IEEE, 2021, arXiv: 2107.01711.
15. G. Williams, R. Baxter, H. He, S. Hawkins, and L. Gu, "A comparative study of RNN for outlier detection in data mining", IEEE International Conference on Data Mining, 2002, pp. 709–712.
16. H. Sarvari, C. Domeniconi, B. Prenkaj, and G. Stilo, "Unsupervised Boosting-Based Auto encoder Ensembles for Outlier Detection", Advances in Knowledge Discovery and Data Mining: 25th Pacific Asia Conference, PAKDD 2021, Virtual Event, May 11–14, 2021, Proceedings, Part I. Cham: Springer International Publishing, 2021.
17. J. Chen, S. Sathe, C. Aggarwal, and S. D. Turaga, "Outlier detection with autoencoder ensembles", Proceedings of the 2017 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, 2017.
18. J. Yang, X. Tan, and S. Rahardja, "Outlier detection: How to Select k for k-nearest-neighbors-based outlier detectors", Pattern Recognition Letters, vol. 174, pp. 112-117, 2023.
19. M. B. Markus, H. P. Kriegel, T. N. Raymond and J. Sander, "LOF: identifying density-based local outliers". SIGMOD Rec., vol. 29(2), pp. 93—104, 2000.
20. M. Chen, K. Zhou, D. Liu, "Machine learning based technique for outlier detection and result prediction in combustion diagnostics", Energy, vol. 290. pp. 130218, 2024.
21. M. L. Shyu, S. C. Chen, K. Sarinnapakorn, and L. Chang, "A Novel Anomaly Detection Scheme Based on Principal Component Classifier", Proceedings of International Conference on Data Mining, 2003.
22. M. N. K. Sikder, and F. A. Batarseh, "7 - Outlier detection using AI: a survey", AI Assurance, pp. 231-291, 2023.
23. Q. Hu, Z. Yuan, K. Qin, and J. Zhang, "A novel outlier detection approach based on formal concept analysis", Knowledge Based Systems, vol. 268, pp. 110486, 2023.
24. S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier Detection Using Replicator Neural Networks," Lecture Notes in Computer Science, Springer, 2454, 2002.
25. S. Haykins, "Neural network and learning machines", Prentice Hall India, 2009.
26. S. Jahan and H. D. Qi, "Regularized Multidimensional Scaling with Radial Basis Functions," Journal of Industrial and Management Optimization, vol. 12, pp. 543–563, 2016.
27. S. Jahan, "Discriminant analysis of regularized multidimensional scaling", Numer. Algebra Control Optim., vol. 11(2), pp. 255–267, 2021.
28. S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000.
29. S. Sathe and C. Aggarwal, "LODES: Local density meets spectral outlier detection", SIAM international conference on data mining. Society for Industrial and Applied Mathematics, 2016.
30. T. Kieu, B. Yang, C. Guo and S. J. Christian, "Outlier Detection for Time Series with Recurrent Autoencoder Ensembles", International Joint Conferences on Artificial Intelligence (IJCAI), pp. 2725–2732, 2019.
31. UCI Machine Learning Repository, http://www.ics.uci.edu/ mlearn/MLRepository.html.

Journal of Artificial Intelligence, Machine Learning and Neural Network
ISSN:2799-1172
Vol: 04, No. 06, Oct-Nov 2024
http://journal.hmjournals.com/index.php/JAIMLNN
DOI: https://doi.org/10.55529/jaimlnn.46.12.26

32. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey", ACM Comput. Surv., vol. 41, 2009.
33. Y. Ma, P. Zhang, Y. Cao, and L. Guo, "Parallel auto-encoder for efficient outlier detection," IEEE International Conference on Big Data, Silicon Valley, CA, USA, pp. 15–17, 2013.
34. Y. Qiao, X. Cui, P. Jin, and W. Zhang, "Fast outlier detection for high-dimensional data of wireless sensor networks", International Journal of Distributed Sensor Networks, vol. 16(10), pp. 1–13, 2020.
35. Y. Singh, and A. S. Chauhan, "Neural Networks in Data Mining", Journal of Theoretical & Applied Information Technology vol. 5(1), pp. 37–42, 2009.
36. Y. wang, Y. Hongxun, and S. Zhao, "Autoencoder based dimensionality reduction", Neurocomputing, vol. 184, pp. 232–242, 2016.