World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(RESEARCH ARTICLE)

Check for updates

Identifying rev model based on online streaming data using deep learning technique

Mohammed Sadhik Shaik *

Sr. Software web developer Engineer, Computer science, Germania Insurance, Melissa, Texas.

World Journal of Advanced Research and Reviews, 2024, 21(02), 2027-2034

Publication history: Received on 16 January 2024; revised on 24 February 2024; accepted on 28 February 2024

Article DOI: https://doi.org/10.30574/wjarr.2024.21.2.0411

Abstract

Web mining has recently grown in popularity as a practical innovation that helps individuals and organizations learn from each other's discoveries. When consumers are in the market for a product, mining helps them narrow their options by focusing on compressed sentiments rather than wasting time on lengthy surveys and drawing out their own plans. Content from surveys is a vital resource for online businesses before customers buy anything. This review covers the history, current state, and potential future of sentiment mining and investigation of review spam. Reader opinion can be swayed by any review, whether it's positive or negative. Consequently, our endeavor employs an automated system for review verification and spam detection. A user can use the proposed DLS-Rev Model to detect and eliminate bogus reviews. The research shows a very trustworthy framework that can detect fake content with 100% accuracy and a 100% extraction rate. The highly pleasing result was obtained by extracting sentiment- and opinion-approved phrases. The project's overarching objective is to make the online review system better for everyone by reducing the negative impact of spammers.

Keywords: Text Mining; Review; Machine learning; Extraction; Accuracy; Spam

1. Introduction

There is an explosion of data due to the proliferation of internet connections and the dramatic rise in internet usage throughout the last decade. "Streaming data" describes this category of big data, which consists of an endless stream of data points organized chronologically from earliest to latest [1]. For that reason, data points in streams are not static but rather constantly changing. Due to memory constraints and data streams' potentially unlimited size, time window models are commonly used for processing.

When Windows is data-driven, it makes an incremental change estimate every 100 iterations; when it's time-based, it makes an estimate, say, every 20 seconds. Most timing windows fall into one of three categories: damped, sliding, or landmark [2]. In the sliding window paradigm, every window starts at the same fixed size, w, regardless of the time t. What this implies is that every data point in the current window is equally relevant, and that each window selectively removes older data points while keeping the most recent ones.

The landmark window model, on the other hand, takes input going from a time-stamp named landmark all the way up to the present moment and processes it by keeping one of its boundaries constant at a specific point in time while letting the other follow the progression of time. However, in a damped window model, the data in the stream is given weights according to its arrival time. The weights of current data are larger than those of previous data, and these weights gradually decrease exponentially over time according to an ageing function. Due to the massive volume of data involved, streaming data is susceptible to outliers, or aberrant data points. By "an observation that deviates so significantly from other observations as to arouse suspicion that it was generated by a different mechanism," Hawkins [3] meant an outlier. These records may be called outliers, anomalies, or deviants in the data mining and statistics literature [4]. It is

* Corresponding author: Mohammed Sadhik Shaik

standard practice to search for data outliers in order to uncover new opportunities and early indicators of risk. The goal of identifying outliers is to locate data points that deviate significantly from the norm. Outlier identification over data streams is becoming more popular as a result of its many practical applications. The following are some examples of potential use cases: healthcare fraud detection, industrial fault detection, early sickness identification in banking and finance, and abnormal vital signs detection. Since the data's underlying distribution may not be known in advance, outlier detection in data streams presents a formidable challenge.

So, this process can be carried out offline as well as online when dealing with streaming data. Offline batch processing, which involves analyzing a large amount of data all at once, might help us spot anomalies in the past. However, as fresh data points are received and outliers are discovered, what is known as real-time processing occurs online. As soon as the data is received, it is processed instantly. Using real-time streaming processing, this study employs the sliding window layout. Deep neural networks (DNNs) have lately emerged as the preferred method for addressing many problems in various domains, including computer vision, network security, natural language processing (NLP), but not limited to.

Their deep architecture allows them to train complex features with many representations, giving them a leg up in these domains. Furthermore, DNN's deep architecture solves the problems with scalability, manual feature engineering, and generalization to new data changes that plague traditional ML methods. That is why DNN is a good method for detecting outliers because of these features [5, 6]. The model is trained using a multi-layer supervised learning method called a deep neural network. How many layers are used to describe the data feature extraction process is what the word "deep" means. Figure 1 shows the fundamental design of a DNN. In addition to an input layer, it has multiple hidden levels and an output layer. The building blocks of several well-known neural networks are layers of neurons; these include CNNs, RNNs, and MLPs (multilayer perceptron networks). Layers like these might be stacked, convolutional, recurrent, or fully linked. Expressive representations are provided by deep neural networks (DNNs) through the use of intricate combinations of computational graph-representable linear and nonlinear functions. "Activation functions" describe these types of functions because they determine the output of computational graph nodes (like neurons in a neural network) in response to specific inputs. A few popular activation functions are sigmoid, linear, Rectified Linear Unit (ReLU), and tanh.

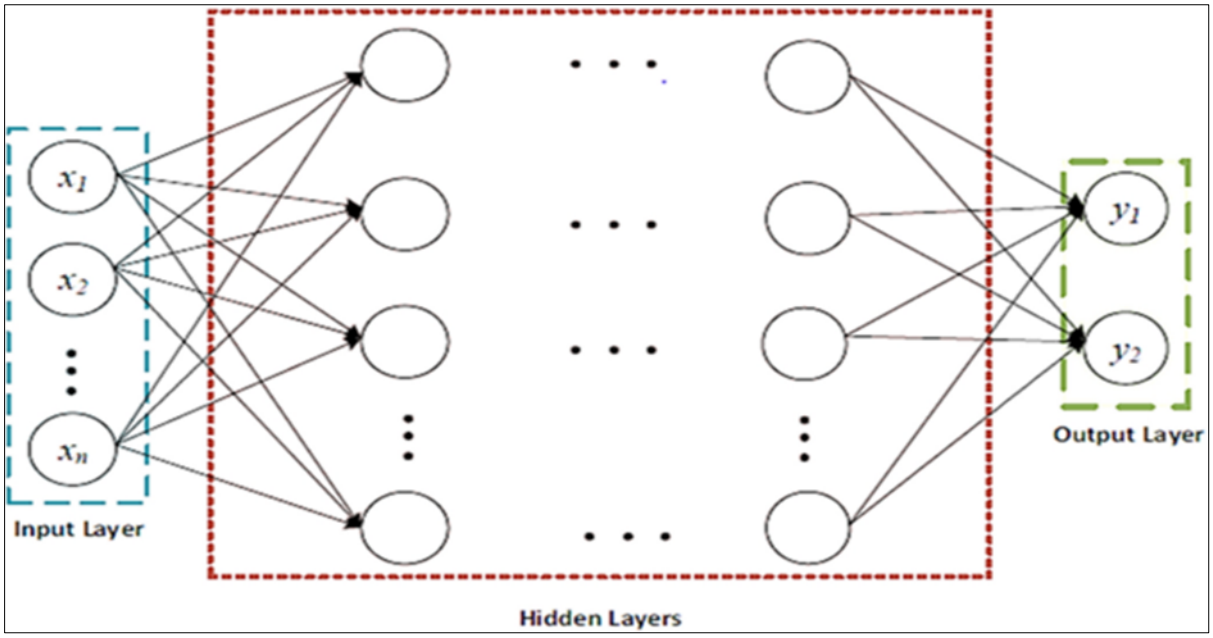

Figure 1 The Deep Neural Network Architecture

An innovative deep learning-based approach for outlier detection in streaming data is introduced in this research, which makes use of a deep neural network (DNN). To find outliers in real-time streaming data, the suggested model is an online system based on DNNs that use the sliding window method. According to the suggested model, the DNN consists of an input layer, three hidden layers, and an output layer that uses two neurons to label occurrences as either outliers or inliers.

2. Related work

There are three main schools of thought when it comes to detecting outliers in data streams: ML, DL, and hybrid methods that incorporate elements of both schools [7]. Machine learning, as seen in Figure 2, allows computers to acquire new knowledge automatically, without human intervention. In doing so, it is possible for the system to draw on both historical and real-time data in order to make choices or predictions about the future [8]. On the other hand, DL is a branch of AI that deals specifically with machine learning (ML).

AI is concerned with programming computers to mimic human intellect and the creation of neural networks for supervised, semi-supervised, and unsupervised learning from both organized and unstructured data[9, 10]. It is now possible to significantly improve outlier detection performance using deep learning techniques as compared to conventional ML methods [11]. At its foundation, deep learning for outlier detection is building feature representations or outlier scores using neural networks [12]. Its remarkable learning capabilities for complicated big data models, such as graphs, trajectories, high-dimensional streaming, and temporal-spatial data, were recently on display.

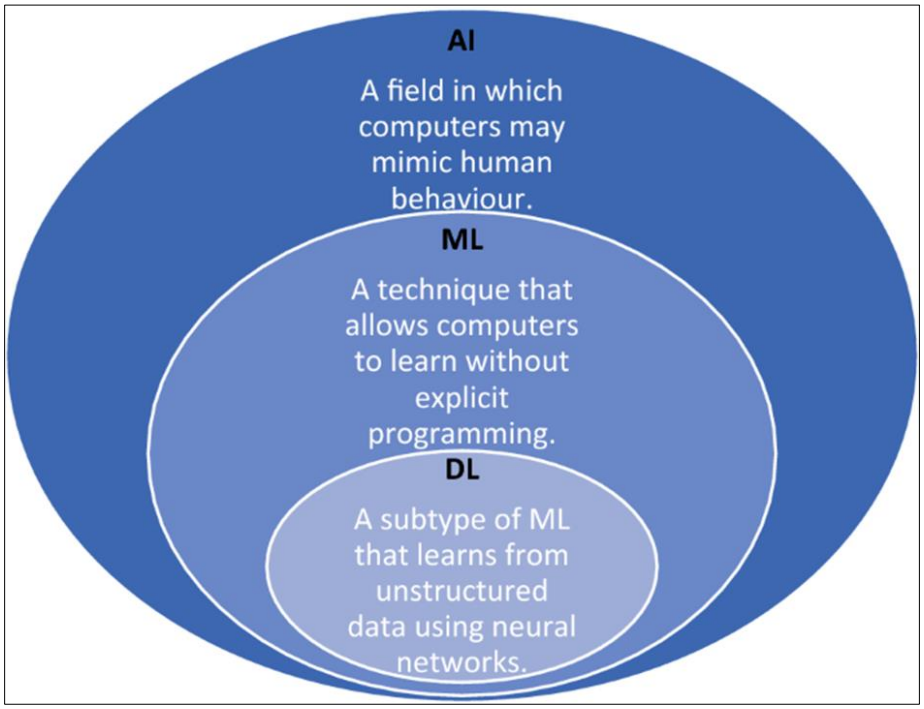

Figure 2 Machine Learning, Deep Learning, and Artificial Intelligence

Many of the current methods for detecting outliers rely on machine learning techniques. Outlier detection for streaming data has been the subject of numerous research methodologies in the literature, including cluster-, distance-, classification-, kernel-, and ensemble-based approaches [14]. However, there have been several new approaches to outlier detection that rely on deep learning. To illustrate the usage of deep neural network activations in a streaming situation, a novel approach to identifying concept evolution was introduced in [15] and dubbed DeepStreamCE. A framework for anomaly identification using DNNs, including explanations for any anomalies found, is suggested in [16], another noteworthy paper. The authors of [17] presented DeepAnT, an unsupervised anomaly detection method that relies on deep learning. It consisted of two parts: one for predicting future timestamps (the time-series predictor module) and another for identifying abnormal data points (the anomaly detector module). In order to train CNN, DeepAnT uses raw data without filtering out outliers. They used a max-pooling layer after two convolutional layers. In order to enhance data for convolutional neural network (CNN) time series anomaly detection, a new article suggests taking advantage of frequency domain amplitude and phase spectra disruptions. In some cases, researchers find that combining ML and DL procedures into hybrid strategies helps them generate better models. Particularly, an ML predictor is examined by each of these methods after DL methods are examined for complexity and feature reduction.

One example is the employment of the encoder component of autoencoder (AE) in a hybrid approach that combines it with the random forest technique [18]. Another hybrid method was developed by Marir et al., who used the ensemble methodology and voting to combine DBN with SVM. In addition, Yan et al. presented an additional hybrid idea that

combined sparse autoencoder with support vector machine (SVM), although this approach failed to identify minority outlier labels. An unsupervised anomaly detection framework called FuseAD was suggested in [19]. It integrates methods from deep learning with statistics. An auto-regressive moving average (ARIMA) algorithm and a convolutional neural network (CNN) were employed by the system to identify outliers in the real-time sensor data.

A new study suggests mixing anomaly detection with federated learning and data streams as a means to further the development of federated learning (FL), a subfield of artificial intelligence and machine learning. Prior work on outlier identification using deep neural networks has not yet achieved high accuracy, precision, recall, and other metrics while minimizing false alarm and miss rates. The outlier detection problem has relied heavily on machine learning techniques for quite some time.

Nevertheless, deep learning outperforms machine learning in this area. Machine learning has numerous shortcomings, such as its inefficiency when it comes to handling new characteristics compared to a preset set of features. But there has been surprisingly little study on using deep learning to spot outliers in streaming data. Researchers have developed effective models for outlier detection using DL approaches such as Deep Belief Networks (DBN), Autoencoders (AE), Generative Adversarial Networks (GAN), Convolutional Neural Networks (CNN), and Deep Neural Networks (DNN) [20].

3. Methodology

3.1. The proposed DNN model

The three-step approach for outlier identification over stream using the proposed DNN model is as follows: data preparation, DNN training, and determination. The proposed DNN-based model's workflow is illustrated in Figure 3. The next sections offer an in-depth examination of the approach that was employed to develop the DNN-based outlier detection model.

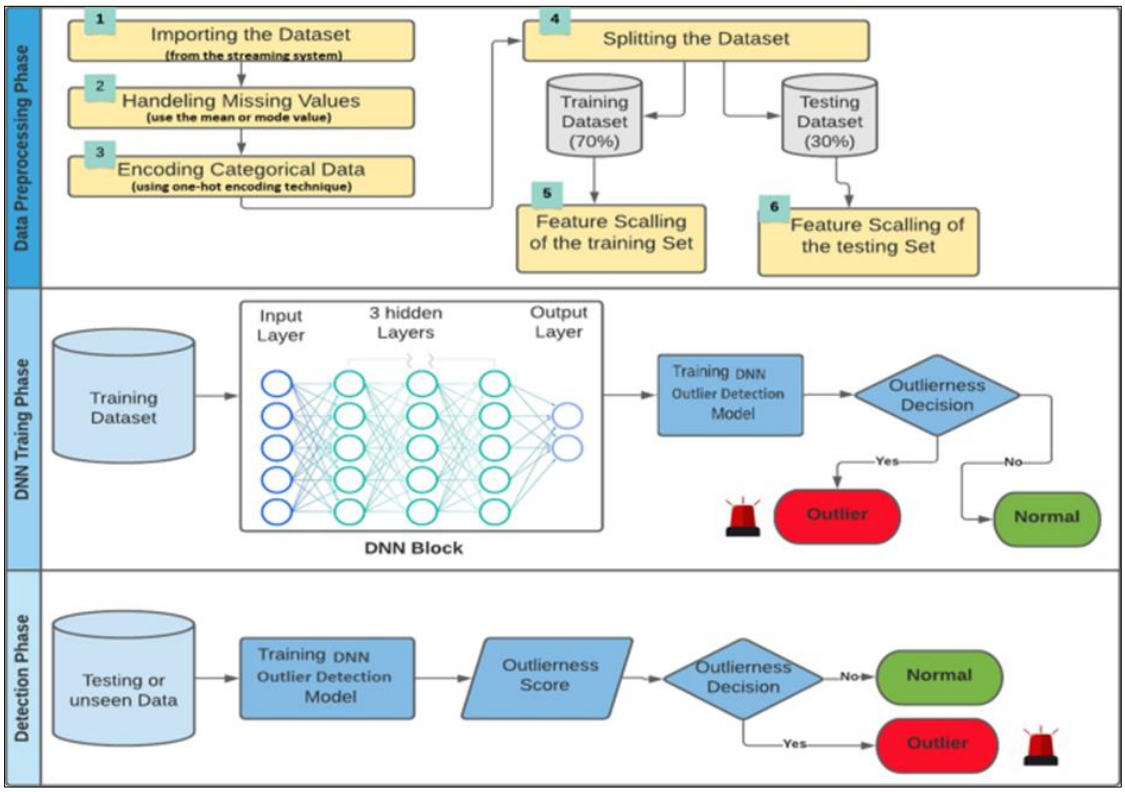

Figure 3 The Outlier Detection Model that is suggested to be built on DNN

3.2. Data preprocessing phase

Data preparation is a crucial step in improving the quality of data, which in turn facilitates the extraction of valuable insights from the data. Outlier detection models typically begin with data preparation, which involves cleaning and preparing raw data for use in creating and training DNN-based models. The five steps that make up this phase are:

- Importing the dataset
- Datasets brought in from the stream systems are used.
- Handling missing values of data

Information technology plays a crucial role in accurately identifying and handling missing values. This is due to the fact that conclusions derived from data can be significantly affected by missing data.

Therefore, if a variable is numerical or categorical and has no values, we shall fill them in accordingly. When numerical data is absent, the variable mean is utilized to fill the gaps. We substitute the variable mode if it's categorical. This eliminates any potential bias caused by missing values in an unbiased manner.

3.2.1. Encoding categorical data

Deep neural networks mostly function on numerical data. Therefore, numerical representations are given to these category variables. For such conversion, the One-Hot Encoding technique is utilized.

4. Results and discussion

Anomaly detection in data streams using the suggested DNN-based model was tested in an experimental setting. For more precision, we used the classification accuracy and the previously mentioned criteria to test the model's decision-making capability. Two more DL methods, DeepAnt and RobustTAD., are compared to the suggested DNN-based model.

As the outlier ratio rises, DeepAnt and RobustTAD's detection ability degrades noticeably. But even when faced with challenging datasets like Cardiotocography and Annthyroid, the suggested model maintains its robustness. Figures 5 and 6 also show the results of the performance measures. With accuracy scores of 98.354% for the Breast-Cancer dataset, 98.644% for the Annthyroid dataset, 99.241% for the Musk dataset, and 99.631% for the Cardiotocography dataset, the suggested model surpassed the other two DL techniques.

DeepAnt, on the other hand, had scores of 91.037%, 93.618%, 90.392%, and 90.816%. The suggested model's accuracy in predicting outlier data on the evaluated datasets was 98.579%, 98.853%, 98.903%, and 99.702%, respectively. Also, compared to DeepAnT and RobustTAD, the suggested model scored better on recall, F1-score, and specificity metrics.

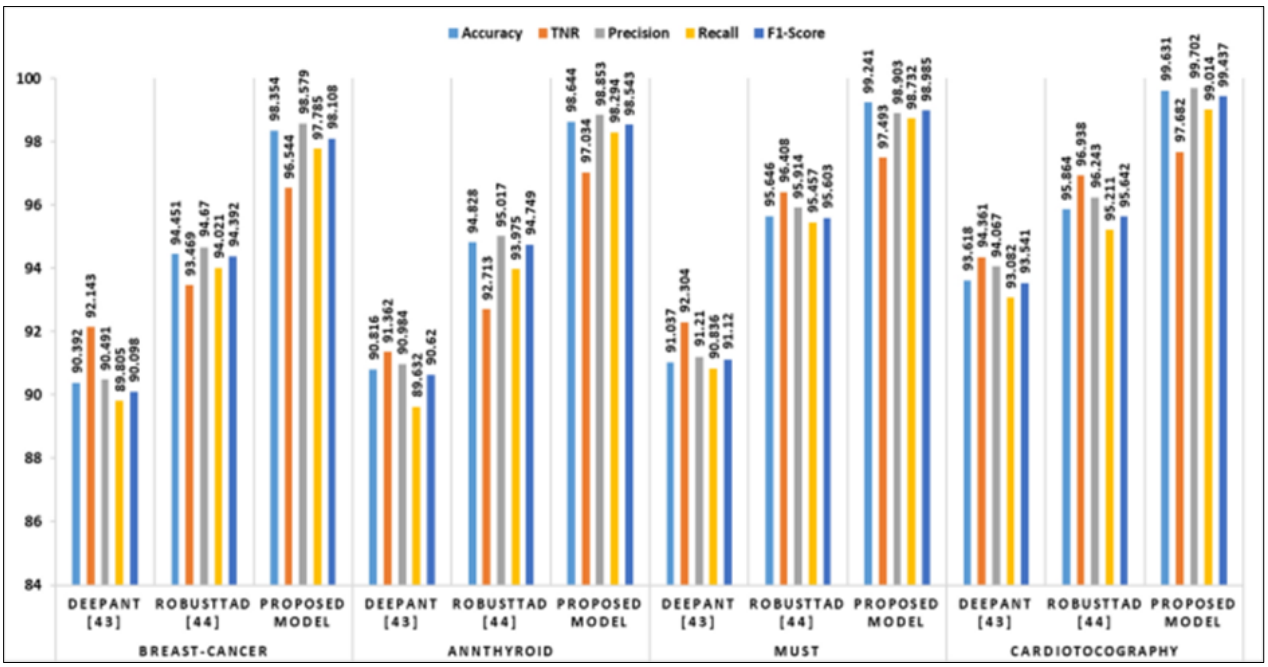

Figure 4 Various DL Methods' Performance Metrics on Various Benchmark Datasets

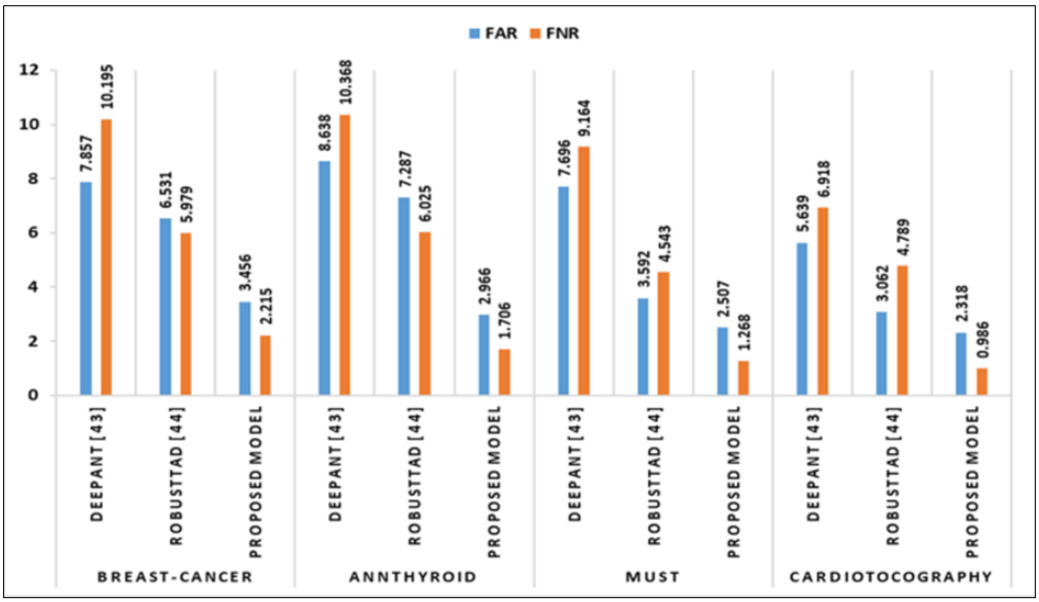

Figure 5 Evaluation Criteria for Various DL Algorithms (FAR and FNR)

Figure 5 displays a comparison of the proposed model's FNR and FAR performance with two other DL-based techniques, namely DeepAnT and RobustTAD. When compared to alternative approaches, the proposed model yielded noticeably reduced FNR and FAR values. Even though it maintained an extremely high detection rate, the false alarm rates were 3.456%, 2.966%, 2.50%, and 2.318%, respectively. The competing models clearly do a poor job of detecting outlier data, since their FNR and FAR rates are higher. The proposed approach outperformed the other two on more complicated, higher-dimensional datasets, in contrast to the simpler, lower-dimensional datasets, such as the Breast-Cancer dataset. Our suggested model outperforms the state-of-the-art alternatives on two datasets—Musk, which contains the most features, and Annthyroid, which includes the most occurrences. Nevertheless, the suggested model outperforms the state-of-the-art models on both the outlier-light Musk dataset and the outlier-heavy Cardiotocography dataset. The following tests evaluate the suggested model in comparison to popular ML techniques including Random Forest, KNN, and SVM. Table 5 provides an overview of the complete experiment using only the Musk and Annthyroid datasets. Due of their extremely high dimensionality, we chose these datasets. The suggested DNN-based model outperforms conventional ML methods in every respect, thanks to the superior performance of DL methods (Fig. 6).

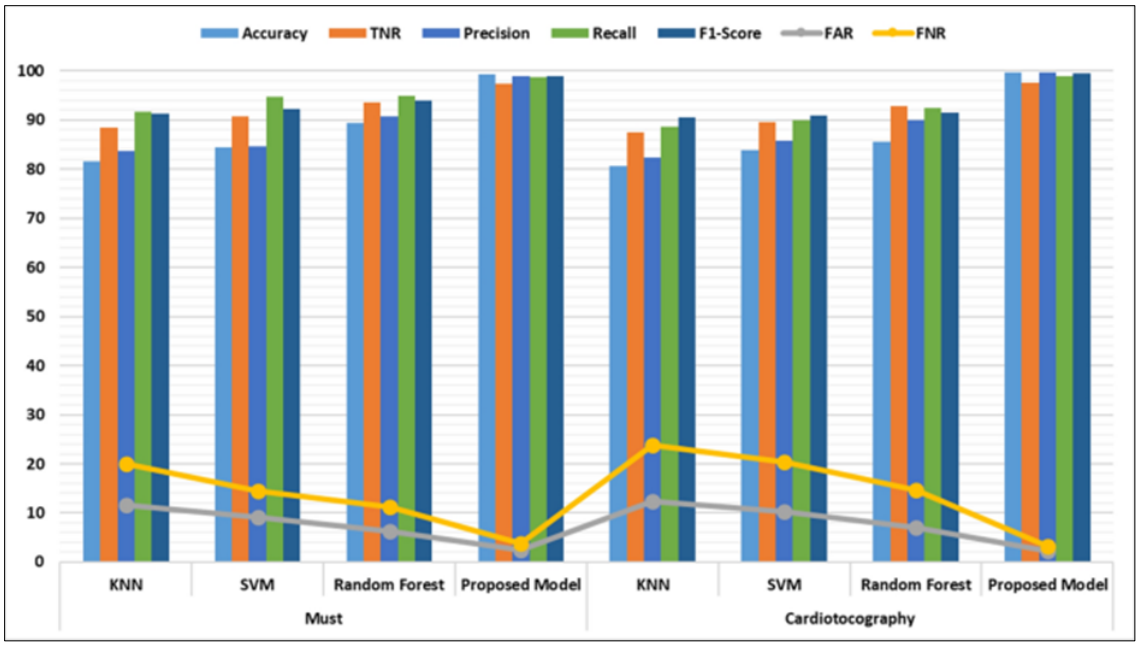

Figure 6 Evaluation of the suggested Model and Other ML Techniques

5. Conclusions and future work

Using deep neural networks (DNNs), this study effectively classifies data flows as either typical or abnormal, presenting a new paradigm for outlier detection in streaming data environments. Experiments utilizing several real-world datasets showed that the suggested model had a false alarm rate of 2.3% to 3.4%, which was lower than two state-of-the-art DL methods. It was also demonstrated that the framework could detect outliers with a recall (or detection rate) of 97-99%. Plus, the proposed model was noticeably better than the conventional ML techniques. However, the current proposed approach isn't totally inadequate because of how long it takes to train the model and how much it focuses solely on finding global outliers. We intend to build upon this work in future iterations by extending it to multiclass classification scenarios and solving the outlier identification issue in data streams with the help of new DL approaches. Furthermore, we want to address the contextual outlier issue.

References

[1] Kim T, Park CH. Anomaly pattern detection for streaming data. Expert Syst Appl. 2020;149:113252. https://doi.org/10.1016/j.eswa.2020.113252.

[2] Mansalis S, Ntoutsi E, Pelekis N, Theodoridis Y. An evaluation of data stream clustering algorithms. Stat Anal Data Min. 2018;11(4):167–87. https://doi.org/10.1002/sam.11380.

[3] Hawkins DM. Identification of outliers, vol. 11. Dordrecht: Springer; 1980.

[4] Aggarwal CC. An Introduction to Outlier Analysis. In: Aggarwal CC, editor. Outlier Analysis. Cham: Springer International Publishing; 2017. p. 1–34. https://doi.org/10.1007/978-3-319-47578-3_1.

[5] Nguyen G, et al. Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. Artif Intell Rev. 2019;52(1):77–124. https://doi.org/10.1007/s10462-018-09679-z.

[6] Czum JM. Dive into deep learning. J Am Coll Radiol. 2020;17(5):637–8. https://doi.org/10.1016/j.jacr.2020.02.005.

[7] Al-amri R, Murugesan RK, Man M, Abdulateef AF, Al-Sharafi MA, Alkahtani AA. A review of machine learning and deep learning techniques for anomaly detection in iot data. Appl Sci. 2021;11(12):5320. https://doi.org/10.3390/app11125320.

[8] Gomes HM, Read J, Bifet A, Barddal JP, Gama J. Machine learning for streaming data: state of the art, challenges, and opportunities. SIGKDD Explor Newsl. 2019;21(2):6–22. https://doi.org/10.1145/3373464.3373470.

[9] Zhang A, Lipton ZC, Li M, Smola AJ. Dive into deep learning, arXiv Prepr. arXiv2106.11342, 2021.

[10] Vargas R, Mosavi A, Ruiz R. Deep Learning: A Review. Adv Intell Syst Comput. 2018. https://doi.org/10.20944/preprints201810.0218.v1.

[11] Pang G, Shen C, Cao L, Van Den Hengel A. Deep learning for anomaly detection. ACM Comput Surv. 2021;54(2):1–38. https://doi.org/10.1145/3439950.

[12] Xue F, Yan W, Wang T, Huang H, Feng B. Deep anomaly detection for industrial systems: a case study. Annu Conf PHM Soc. 2020;12(1):8. https://doi.org/10.36001/phmconf.2020.v12i1.1186.

[13] Cao F, Estert M, Qian W, Zhou A. Density-based clustering over an evolving data stream with noise, in Proceedings of the 2006 SIAM International Conference on Data Mining, Apr. 2006;2006:328–339. https://doi.org/10.1137/1.9781611972764.29.

[14] Constantinou V. PyNomaly: anomaly detection using local outlier probabilities (LoOP). J Open Source Softw. 2018;3(30):845. https://doi.org/10.21105/joss.00845.

[15] Yang X, Zhou W, Shu N, Zhang H. A Fast and Efficient Local Outlier Detection in Data Streams, in Proceedings of the 2019 International Conference on Image, Video and Signal Processing, 2019;111–116. doi: https://doi.org/10.1145/3317640.3317653.

[16] Huang JW, Zhong MX, Jaysawal BP. Tadilof: time aware density-based incremental local outlier detection in data streams. Sensors. 2020;20(20):1–25. https://doi.org/10.3390/s20205829.

[17] Singh M, Pamula R. ADINOF: adaptive density summarizing incremental natural outlier detection in data stream. Neural Comput Appl. 2021;33(15):9607–23. https://doi.org/10.1007/s00521-021-05725-0.

[18] Abid A, El Khediri S, Kachouri A. Improved approaches for density-based outlier detection in wireless sensor networks. Computing. 2021;103(10):2275–92. https://doi.org/10.1007/s00607-021-00939-5.

[19] Hassan A, Mokhtar H, Hegazy O. A heuristic approach for sensor network outlier detection. Int J Res Rev Wirel Sens Netw. 2011;1(4):66–72.

[20] Fawzy A, Mokhtar HMO, Hegazy O. Outliers detection and classification in wireless sensor networks. Egypt Informatics J. 2013;14(2):157–64. https://doi.org/10.1016/j.eij.2013.06.001.