CENTROID STABILIZED FUZZY TUKEY QUARTILE AND Z CURVE NEURAL NETWORK BASED OUTLIER DETECTION

S. RAJALAKSHMI[1]
P. MADHUBALA[2]

[1]Department of Computer Science, Research Scholar, Periyar University, Salem, India
[2]Department of Computer Science, Research Supervisor, Periyar University, Salem, India
[1]rajaylakshmiravi7@gmail.com
[2]madhubalasivaji@gmail.com

Abstract. This paper presents a new method called, Centroid Stabilized Tukey Quartile and Z-curve Neural Network (CSTQ-ZNN) for outlier detection. The main purpose of this paper is to investigate numerous factors that hide outliers and produce the best clustering results via noise reduction, perpetual outlier identification, and centroid stabilization. Moreover unusual outliers are identified using fuzzy clustering and improve computational efficiency by means of centroid stabilization via Tukey Quartile function. The CSTQ-ZNN method is split into two sections. They are clustering of the data points and outlier detection. First Centroid Stabilized Fuzzy Tukey Quartile-based Clustering model is applied to the raw NIFT-50 Stock Market Dataset. Next, with the processed clusters as input, Deep Z-Curve Neural Network model is presented for outlier detection. In contrast, after an analysis of comprehensive experiments performed to validate the CSTQ-ZNN method via comparisons against existing methods and benchmark performance metrics, we found that our proposed method performs better than existing methods in terms of time complexity, error tolerance, true negative rate and outlier detection accuracy.

Keywords: Centroid Stabilization, Tukey Quartile, Z-curve, Deep Neural Network, Outlier Detection

(Received September 21st, 2022 / Accepted December 11nd, 2022)

1 Introduction

An outlying scrutiny, or outlier, is one that emerges to diverge noticeably from other members of the sample in which it takes place. Their detection can recognize system faults and fraudulent activities before they shoot up with possibly disastrous repercussions. As outliers are fascinating due to the reason that they are reckoned of not being given rise by the same procedures as the rest of the data, it is significant to account for why detected outliers are given rise by certain other mechanisms.

Angiulli et al. [3] introduces the Kernel Density Estimation (KDE) to address the issue of explanation and detection of anomalous values in categorical datasets. The value of the attribute was perceived in such a manner to be of anomalous if its frequency was observed to be exceptional within overall frequency distribution. Initially, the concept of frequency occurrence were elaborated and then applied to the domain of frequency values. Next, an outlierness estimate for categorical values was obtained via cumulated frequency distribution. Here, two types of anomalies, namely, lower outliers and upper outliers were measured. With this the method was found to be scalable and was also successful in identifying anomalies in a computationally efficient manner. However, the time complexity factor was not analyzed.

An Iterative ensemble method with distance-based data filtering integrating two hard clustering algorithms and one soft clustering algorithm was presented by Chakraborty et al[7]. Here, the bias was eliminated whereas agility in terms of robust search space was introduced. Also, Dunn index based threshold criterion was introduced that in turn ensured progressive outlier search. Moreover for addressing issues arising out of class imbalance, an intelligent learning step was introduced. With this ensemble pattern precision in addition to F-score were improved. Despite improvement in F-score value, the outlier detection accuracy was not focused.

A review of deep learning technique for detecting anomalies concerning time series data was investigated by Choi et al [8]. Despite cluster based outlier detection techniques detect outliers from static data however arbitrary data presence necessitates detection operation in a single pass manner. Moreover, the cluster based outlier detection techniques does not take into consideration the time correlation factor between arbitrary data that in turn compromises the detection accuracy rate. To address on this issue, an effective outlier detection technique was designed by Cai et al.[6] on the basis of neighbor difference and clustering, called, Outlier Detection based on Neighbor Difference and Clustering (ODNDC). This in turn not only detected outlier in an accurate manner but also identified the outlier source. Though outlier source were obtained in a significant manner, the error tolerance was not focused.

Dutta et al.[10] develops an outlier detection model based on rarity in a sparse coding framework. The framework was split into two sections. They were, Negative Log Activity Ratio (NLAR) learning and outlier scoring. The algorithm was designed in an unsupervised fashion and both the offline and online variants were presented that in turn operated in linear time. Yuan et al.[22] investigates the clustering method to detect outliers. However, the true negative rate was not focused.

We introduce Centroid Stabilized Tukey Quartile and Z-curve Neural Network (CSTQ-ZNN) for outlier detection to quantify the outliers of the data point for this problem, and address the issue of not having adequate a priori knowledge and labeled data in stock market outlier detection.

The major contributions of this work are concluded as follows:

- With the use of extreme limits, Tukey Quartile and Centroid Stabilization function, we propose an efficient clustering algorithm to obtain time efficient and error tolerance-based clusters with the consideration of time correlation of the data, and then use the w-kmeans algorithm with the consideration of dual pair-wise stock market data points and mutation function to reduce the impact of noise on the clustering results, thereby improving the detecting performance.
- With the consideration of maxout function and Z-score, we propose an efficient outlier source identification algorithm called, Deep Z-Curve Neural Network-based Outlier Detection to improve the outlier detection with minimum true negative rate.
- Based on a NIFT-50 Stock Market Dataset, we conduct extensive experiments to evaluate the efficiency of the method, Centroid Stabilized Tukey Quartile and Z-curve Neural Network (CSTQ-ZNN) for outlier detection, and the experimental result verifies that the proposed CSTQ-ZNN method can accurately detect potential outliers from stock data as well as minimize time with high error tolerance rate.

The remaining of this paper is organized as follows. Section 2 presents the related works. Section 3 first describes the framework of the proposed method for outlier detection, and then presents the details of the proposed approach. Section 4 and 5 demonstrates our experimental results and discussion. Finally, we conclude our paper in Section 6.

2 Related Works

A novel unsupervised learning method employing Mixed Integer Optimization mechanisms for generating interpretable tree-based clustering models was proposed by Bertsimas et al. [4]. The learning method with the aid of optimization-driven framework attained globally optimal solution resulting in high quality partitions of feature space. Yet another outlier detection method employing novel evolutionary clustering algorithm concentrating on data stream behaviors and adapting accordingly was presented by Nordahl et al.[15].

Most of the prevailing outlier detection methods based on the dynamic graph embedding structure specifically concentrated on the graph evolution whereas discarded the similarities between them. To overcome this disadvantage, a dynamic graph embedding method on the basis of graph proximity, called DynGPE was presented by Li et al. [13]. Here, the events corresponding to different climatic conditions were denoted in the form of graph, with vertex representing the meteorological data and edge denoting relationship between meteorological time series data. This in turn resulted in the improvement of F-measure and accuracy. However, the categorical representation of data was not included. To address on this aspect, an algorithm based on rough set was designed by Salem et al.[16] that with the aid of Rough K-modes formulated categorical representation of clusters with better accuracy.

As far as outlier detection is concerned, clustering mechanism plays a major role with which normal patterns are said to be clustered in a group and abnormal patterns are clustered in another group for discarding from further processing. Vasconcelos et al. [17] analyzes the complex event processing by employing an online k-means algorithm. Yet another density peak algorithm was designed by Zhou et al. [23] that initially evaluated the distance between distance points and cut off distance, therefore ensuring robustness and effectiveness.

A randomized method that initially converts high dimensional database to a binarized form utilizing projected samples of original database was proposed by Moens et al. [14]. Moroever, this database was utilized in mining frequent itemsets that in turn was found to be translated back to subspace clusters. In this manner, several multiple subspaces were said to be explored with distinct sizes concurrently. Variability in the centroid was utilized by Bushra et al. [5] for observing spatiotemporal trends. Wei et al. [19] examined the large-scale outlier detection methods concerning environmental factors by employing machine learning k-means clustering. With this the outlier detection rate were found to be improved. Yet another self supervised learning based cluster was designed by Diers and Pigorsch [9] for outlier detection.

Over the past few years, network environment has become increasingly complicated. This is due to the explosive nature of traffic data and hence has had a paramount influence on both the society and the economy. Also owing to the increase in Internet services, network abnormalities has also increased having negative influence on web services and resulting in both social and economic losses.

Feng et al. [11] designed an anomaly detection algorithm based on X-means and iForest algorithm with the objective of improving anomaly detection rate even in the presence of high traffic. Yet another method using unsupervised learning was proposed by Yoshihara and Takahashi [21] for web time series data. Here, ratio of log likelihood was employed that in turn detected outlier in a timely manner. A plethora of deep learning techniques were investigated by Hooshmand and. Hosahalli [12] for anomaly detection. Yanxia et al. [20] discussed to concentrate on the processing time involved in anomaly detection, relative mass and half space tree based method.

An oppositional ant lion optimizer-based feature selection with a machine learning-enabled classification (OALOFS-MLC) approach was introduced by Venkateswarlu et al. [18]. But, the outlier detection and data clustering approaches were not considered. A new approach was discussed by Afzal et al. [1] to solve the multivariate outlier detection issue. Convolutional neural network (CNN) was analyzed by Alharbi et al. [2] for improving accuracy. However, the inaccessibility of adequate noise-free data and the essential for a priori knowledge make these clustering based methods still laborious and cumbersome to apply to stock market data set scenarios.

3 Methodology

Outlier detection is a well-known region of research in the mining of data sets. It is an essential task in different application domains such as intrusion detection, mobile phone, and insurance claim fraud detection, medical and public health outlier detection, and industrial damage detection. Outlier Detection refers to the process of identifying the data objects whose characteristics and behavior are distinct from the rest of the data objects in the concerned data set. In this work, a method called Centroid Stabilized Tukey Quartile and Z-curve Neural Network (CSTQ-ZNN) for outlier detection is designed. Fig.1 shows the architecture of the CSTQ-ZNN method.

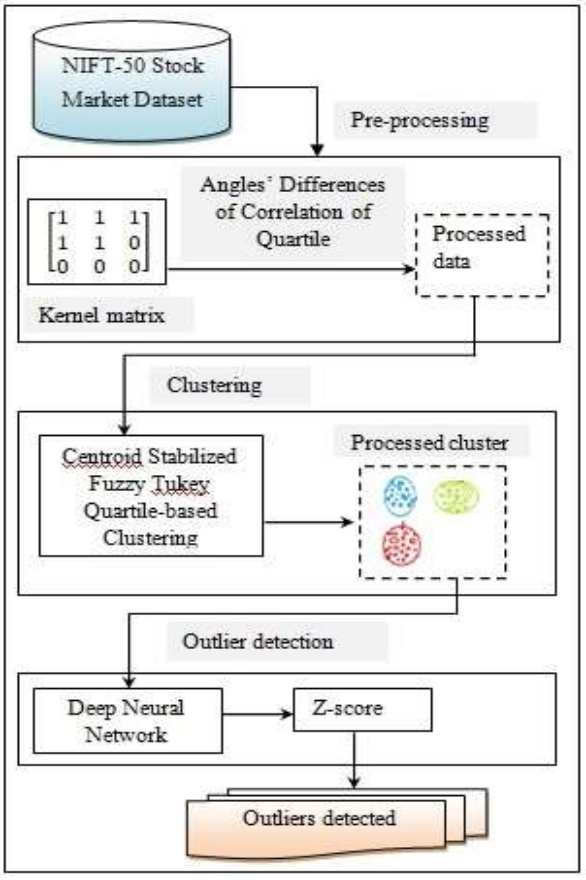

Figure. 1 Architecture of Centroid Stabilized Tukey Quartile and Z-curve Neural Network

As shown in the above figure. 1, with the NIFT-50 Stock Market Dataset provided as input, the proposed Centroid Stabilized Tukey Quartile and Z-curve Neural Network method for outlier detection is split into three major phases. In the first phase, a preliminary pre-processing via kernel matrix is conducted in order to obtain the main data points on which the algorithm works, i.e., outlier detection. In the second phase, a Centroid Stabilized Fuzzy Tukey Quartile-based Clustering model is carried out in order to recognize centroid stabilized processed clusters for further processing. Finally, the actual outlier detection is performed by utilizing Deep Z-Curve Neural Network-based Outlier Detection algorithm. The elaborate description of the CSTQ-ZNN method is provided in the following sections.

Centroid Stabilized Fuzzy Tukey Quartile-based Clustering

The first step towards the proposed method remains in estimating the distinct factors that hide outliers. Therefore, outlier detection initially pre-processes the given raw dataset. This is owing to the reason that with the presence of noise, inconsistence or redundant data, detecting outliers remains a computationally costly process. However, when the data is corrupted or polluted with a blend of distinct noise distributions, the time complexity or the time involved in outlier detection also increases. Also, with the selection of centroids in a random manner minimizes the cluster quality.

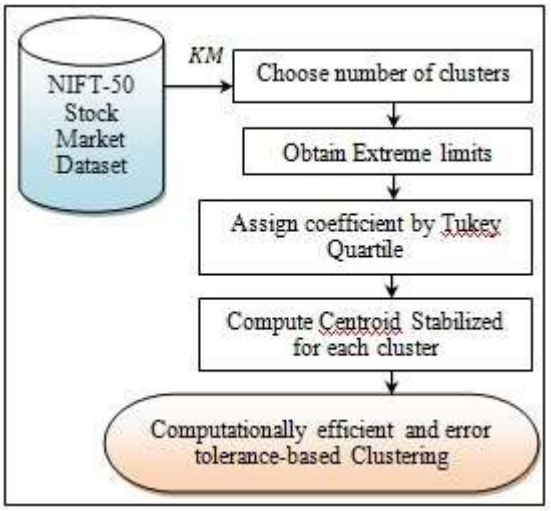

Figure. 2 Flow diagram of Centroid Stabilized Fuzzy Tukey Quartile-based Clustering model

To address these issues, i.e., improving accuracy levels and minimize computational complexity, in our work, initially pre-processing of raw data from NIFTY-50 Stock Market Data (2000 - 2021) is performed followed by which robust clusters are formed by employing Centroid Stabilized Fuzzy Tukey Quartile-based Clustering model. Figure. 2 shows the flow diagram of Centroid Stabilized Fuzzy Tukey Quartile-based Clustering model.

As given in the above figure. 2, with the NIFT-50 Stock Market Dataset provided as input, the objective remains in designing a fuzzy-based clustering model to reduce complexity involved and ensure error tolerance for enabling enable detection and ignoring of outliers. To start with the number of clusters is initialized, followed by which, two extreme limits, i.e., upper and lower limits are identified based on the data points.

Next, coefficients are assigned to each data point by employing Tukey Quartile function. Finally, by means of Centroid Stabilization function for computing centroid for each cluster, error tolerance is ensured in a computationally efficient manner. To start with a kernel matrix is introduced for analyzing cluster distribution. The kernel matrix employed in our work is defined as

given below.

$$KM = \begin{bmatrix} km_{11} & km_{12} & km_{13} & \dots & km_{1n} \\ km_{21} & km_{22} & km_{23} & \dots & km_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ km_{m1} & km_{m2} & km_{m3} & \dots & km_{mn} \end{bmatrix} \quad (1)$$

From the above kernel matrix (1), 'KM' represent the kernel function of the corresponding pair-wise stock market data points. Here, 'm' represents the rows (i.e., stock price data of the fifty stocks in NIFTY-50 index from NSE India) and 'n' represents the columns (i.e., features or certain macro-information about the stocks itself). The above kernel matrix is employed in evaluating the similarities between pair-wise stock market data points, and is defined as given below.

$$km(P_i, P_j) = \varphi\,(P_i)^T . \varphi(P_j),$$
$$where\ i = 1,2, \dots, m\ \&\ j = 1,2, \dots, n \quad (2)$$

From the above equation (2), 'P_i' and 'P_j' represents the dual pair-wise stock market data points and 'φ' denotes the mutation function that predicts the pair-wise stock market data points into a propagating kernel space. Let us consider a dataset 'DS^N' reconstructed from 'N' dimension to multivariate distance space by utilizing a distance metric based on Angles' Differences of the Correlation of Quartile. This is performed by estimating the distance of each observation (i.e., stock market data) in 'N' dimensional space to its Correlation of Quartile function.

With multivariate Angles' Differences of the Correlation of Quartile, possessing five criterions designated as, the Upper Extreme Limits (UEL), the Lower Extreme Limits (LEL), the Upper Quartile (UQ), the Lower Quartile (LQ) and the Median Quartile (MQ) respectively. Then, the dataset 'DS^N' utilized for measuring the extreme values are formulated as given below.

$$LEL_{Dis} = Q1_{Dis} - c_1(Q2_{Dis} - Q1_{Dis}) \quad (3)$$
$$UEL_{Dis} = Q3_{Dis} - c_2(Q3_{Dis} - Q2_{Dis}) \quad (4)$$

From the above two equations (3) and (4), the extreme values are identified based on the data points those lie outside the limits of 'LEL_{Dis}' and 'UEL_{Dis}' respectively. Moreover, the constants, 'c_1' and 'c_2' are variable that changes according to the collected trading volumes. Let us further consider '$P = \{p_1, p_2, \dots, p_N\}$', with '$p_i \in DS^N$', '$N$' representing the dimension as mentioned above and suppose 'P' is said to be allocated to 'c' cluster centers '$cc = \{cc_1, cc_2, \dots, cc_c\}$'.

The objective of the Fuzzy Tukey Quartile-based Clustering is given below where '$k > 1$' denotes the fuzzifier and 'μ_{ij}' represents the fuzzy membership of 'p_i' to cluster center 'cc_j' is mathematically formulated as given below.

$$J = \sum_{i=1}^{N} \sum_{j=1}^{c} \mu_{ij}^{k} \left(p_{ij} - cc_j\right)^2 \quad (5)$$

The updating rule of the membership 'μ_{ij}' and the cluster center 'cc_j' based on the Angles' Differences of the Correlation of Quartile are given below. The iteration of Fuzzy Tukey Quartile-based Clustering is to update fuzzy memberships 'μ_{ij}' and the cluster centers 'cc_j' alternately until convergence.

$$X = \tan^{-1}\left(\frac{\mathrm{p_i}}{\mathrm{q_i}}\right) \{UEL_{Dis}\} \quad (6)$$
$$Y = \tan^{-1}\left(\frac{\mathrm{q_i}}{\mathrm{p_i}}\right) \{LEL_{Dis}\} \quad (7)$$
$$\mu_{ij} = X * Y \quad (8)$$
$$cc_j = \frac{\sum_{i=1}^{N} \mu_{ij}^{k} p_i}{\sum_{i=1}^{N} \mu_{ij}^{k}} \quad (9)$$

From the above equations (6), (7) and (8), fuzzy memberships and the cluster centers are updated based on the two stock market data vectors '$\mathrm{p_i}$', '$\mathrm{q_i}$' respectively. Finally, centroid stabilization (i.e., centroid shift map) is evaluated by employing the individual shift maps as given below.

$$CSM_p(p, q) = \frac{\sum_{CI=1}^{l} ISM_p(p,q,CI)}{CI} \quad (10)$$
$$CSM_q(p, q) = \frac{\sum_{CI=1}^{l} ISM_q(p,q,CI)}{CI} \quad (11)$$

From the above equations (10) and (11), the centroid shift map 'CSM_p', 'CSM_q' for two stock market data vectors '$\mathrm{p_i}$', '$\mathrm{q_i}$', is obtained based on the individual shift map 'ISM_p', 'ISM_q' and cluster index 'CI' respectively. The pseudo code representation of Centroid Stabilized Fuzzy Tukey Quartile-based Clustering is given below.

Input: Dataset 'DS', Features '$F = \{f_1, f_2, \dots, f_n\}$'
Output: Time efficient and error tolerance-based clustering
1: **Initialize** rows '$m = 50$', columns '$n = 15$', '$c_1 = 1.5$', '$c_2 = 1.5$', cluster index 'CI' 2: **Begin** 3: **For** each Dataset 'DS' with Features 'F' 4: Choose number of clusters 'CI' 5: Formulate kernel matrix for analyzing cluster distribution as given in equation (1) 6: Formulate dual pair-wise stock market data points and mutation function as given in equation (2) 7: Evaluate extreme values based on data points that lie outside the limits as given in equation (3) and equation (4) 8: Formulate objective function as given in equation (5) 9: **For** each stock market data vectors '$\mathrm{p_i}$', '$\mathrm{q_i}$' 10: **Repeat** 11: Update fuzzy memberships and the cluster centers as given in equations (6), (7), (8) and (9) 12: Measure centroid stabilization as given in equations (10) and (11) 13: Evaluate the coefficients of being in the cluster 14: **Until** convergence 15: **End for**

16: **Return** processed clusters 'PCL'
17: **End for**
18: **End**

Algorithm 1 Centroid Stabilized Fuzzy Tukey Quartile-based Clustering algorithm

As given in the above Centroid Stabilized Fuzzy Tukey Quartile-based Clustering algorithm, with inputs obtained from NIFT-50 Stock Market Dataset, initially kernel matrix is generated to store the raw data. Second, with the objective of reducing the noise preprocessing is first done by utilizing Angles' Differences of the Correlation of Quartile function. By applying this function, with the noise reduced data points, the time complexity involved in further processing is said to be reduced. Next, with the objective of minimizing or improving error tolerance, a Centroid Stabilization function instead of random centroid is employed that with the aid of individual shift maps ensures a good measure of how well the data points in the cluster are stabilized and close to each other. The more the mean cluster is sharp, the more the clusters are geometrically aligned. As a result, the error tolerance is also said to be improved during the clustering process.

4 Deep Z-Curve Neural Network-based Outlier Detection

Upon successful generation of clusters, the second step remains in detecting the outliers. Despite clustering accuracy is said to be dependent on data (i.e., data point) purity, optimized clustering is said to be ensured, however, significant outlier detection still remains a major tasks. If not properly detected with outliers will compromise both the true negative rate and also the outliers. Therefore in our wok to address the issues concerning true negative rate and outlier detection Deep Z-Curve Neural Network-based Outlier Detection model is proposed. Deep Learning (DL) concept is one of the impressive challenges in computational biology. DL models are played a key part in handling vital issues such as revolutionized speech recognition, visual object recognition, and object detection in computational biology. DL is a type of machine learning (ML) algorithm. ML approach is a computational method basis on statistics, implemented in software. Fig. 3 shows the block diagram of Deep Z-Curve Neural Network-based Outlier Detection model.

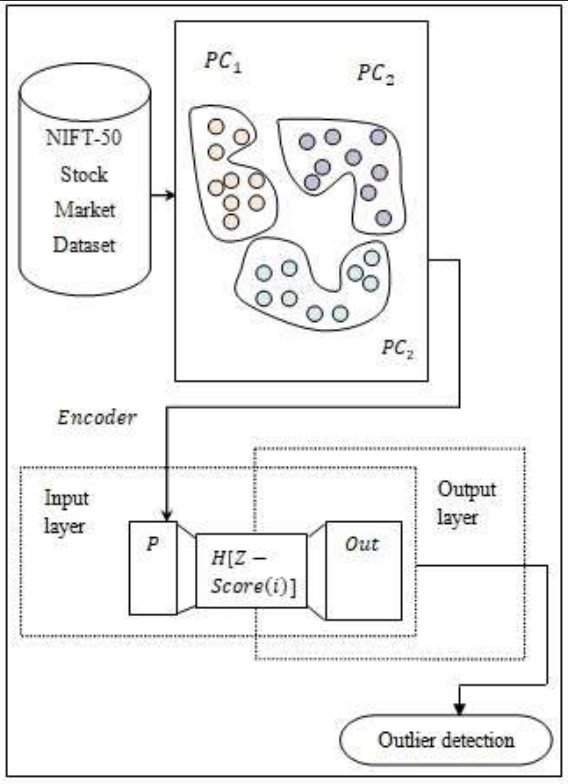

Figure. 3 Block diagram of Deep Z-Curve Neural Network- based Outlier Detection model

As shown in the above figure. 3, auto-encoder being a significant unsupervised neural network model obtains paramount data point representations by means of encoder and decoder. On one hand, the encoder compresses data points into a processed cluster, and on the other hand, the decoder obtains original data points from the encoded features. Finally, with the aid of Z-score measures how far away a certain data point is from the mean and vice versa. Z score is a vital idea in statistics. In addition, the Z score is known as the standard score. Here, the objective remains in separating a core of regular observations (i.e., regular data points) from the outliers (i.e., polluted data points) with higher amount of true negative rate and outliers being detected.

Let us consider 'n' data points '$P = \{p_1, p_2, \dots, p_n\}$' in the processed clusters 'PCL' then the output of '$u-th$' neuron in the '$l-th$' layer of the auto-encoder is denoted as 'Out_{ui}^{l}' and mathematically formulated as given below.

$$Out_{ui}^{l} = f\left(\sum_{j=1}^{n_l}\left[Out_{ji}^{l-1}w_{ju}^{l} + b_u^l\right]\right) \quad (12)$$

From the above equation (12), '$l \in \{1,2,\dots,L\}$' represents the index of layer with 'L' denoting the total number of layers in the auto-encoder. The 'w_{ju}^{l}', 'b_u^l' and 'n_l' represent the weight, bias and number of neurons (i.e., data points) in the '$l-th$' layer. The 'f'

is an activation function and we have utilized the maxout function '$h(p) = (Out_1, Out_2, \dots, Out_n)$' as the activation function. Then, the output vector is represented as 'Out_i^l' and as the first layer is an input layer, it is represented as '$Out_i^1 = p_i$'. In a similar manner, with representation to the decoding process, the decoder maps the deep features '$F = \{f_1, f_2, \dots, f_n\}$' to the reconstruction of data point 'p_i' in the corresponding cluster with multi-layer neurons (i.e., data points). In a similar manner to the output 'Out_{ui}^l' of the '$l - th$' layer, the output of the '$u - th$' neuron in the final layer of the auto-encoder is mathematically formulated as given below.

$$Out_{ui}^l = H_{WB}(p_i)_u = f\left(\sum_{j=1}^{n_L} Out_{ji}^{L-1} W_{ju}^L + b_u^L\right) \quad (13)$$

From the above results of the final layer of the auto-encoder (13) with multi-layer neurons (i.e., data points), the partial derivatives of 'w_{ju}^l' and 'b_u^l' are mathematically represented as given below.

$$\frac{d}{dw_{ju}^l} OF(p, \theta) = Out_{ui}^l W_{ui}^{l-1} \quad (14)$$

$$\frac{d}{db_u^l} OF(p, \theta) = b_{ui}^l b_{ui}^{l-1} \quad (15)$$

From the above equations (14) and (15), the partial derivatives of weight and bias are obtained based on the objective function 'OF' with respect to the numbers of data points 'p' in the corresponding processed clusters. Finally, the presence of outliers are detected by employing the Z-score of any data points 'p' as given below.

$$Z - Score(i) = \frac{\frac{d}{dw_{ju}^l} OF(p,\theta), \frac{d}{db_u^l} OF(p,\theta)[p(i) - mean]}{SD} \quad (16)$$

From the above equation (16) results, the '%' of data points 'p_i' that lie between '$-1/+1\ SD$' is said to be '68%', '$-1/+2\ SD$' is '95%' and '$-1/+3\ SD$' is '99.78%' respectively. Therefore, if the Z-score of any data points 'p' is found to be greater than '3', that data point 'p' is said to be an outlier and vice versa. The pseudo code representation of Deep Z-Curve Neural Network-based Outlier Detection is given below.

Input: Dataset 'DS', Features '$F = \{f_1, f_2, \dots, f_n\}$'
Output: Robust outlier detection
1: **Initialize** processed clusters 'PCL', data points '$P = \{p_1, p_2, \dots, p_n\}$' 2: **Begin** 3: **For** each Dataset 'DS' with Features 'F' and processed clusters 'PCL' 4: **For** each data points 'P' 5: Evaluate output of '$u - th$' neuron in the '$l - th$' layer as given in equation (12) 6: Evaluate output of the '$u - th$' neuron in the final layer as given in equation (13) 7: Evaluate partial derivatives of weight as given in equation (14) 8: Evaluate partial derivatives of bias as given in equation (15) 9: Evaluate Z-score of any data points as given in equation (16) 10: **If** '$Z - Score(i) > 3$' **then** 11: Data point 'p_i' is outlier 12: **Else** 13: Data point 'p_i' is not outlier 14: **End if** 15: **End for** 16: **End for** 17: **End**

Algorithm 2 Deep Z-Curve Neural Network-based Outlier Detection

As given in the above algorithm, the objective remains in detecting the outliers with high true negative rate. With this objective, a deep neural network model employing auto encoder and decoder is designed. Here, the input forms the data points in the processed clusters and the output results in the outliers being detected and vice versa. For detecting the outliers, the affinities between new representations (i.e., from the succeeding and preceding layers) are fine tuned in accordance with the discriminative information by utilizing maxout function. With the application of this function, result in the improvement of true negative rate. Finally, by shifting the discriminative information and making it zero mean with unit standard deviation results in the improvement of outlier detection.

5 Experiments and analysis

In this section, experiments are conducted to verify the validity of the Centroid Stabilized Tukey Quartile and Z-curve Neural Network (CSTQ-ZNN) for outlier detection method using NIFT-50 Stock Market Dataset (https://www.kaggle.com/datasets/rohanrao/nifty50-stock-market-data). All performance metrics are implemented on a Windows 10 operating system computer with an Intel(R) Core (TM) i5-9300H (2.40 GHz) processor, 16 GB of RAM by means of R Programming language.

6 Result

In this section, the experimental results of CSTQ-ZNN method against existing methods using Centroid Stabilized Fuzzy Tukey Quartile-based Clustering algorithm is discussed and test our experimental results in terms of time complexity and error tolerance. Then, comparison statistics for outlier detection are made using Deep Z-Curve Neural Network-based Outlier Detection and test the performance results in terms of true negative rate and outliers being detected.

Case Scenario 1 : Time Complexity

A significant amount of time is said to be consumed during the process of outlier detection. To be more specific, the time complexity is the computational complexity that specifies the amount of computer time it takes to execute the Centroid Stabilized Fuzzy Tukey Quartile-based Clustering algorithm. Time complexity is estimated by counting the number of operations performed by Centroid Stabilized Fuzzy Tukey Quartile-based Clustering algorithm. In our work, the time complexity is expressed as '$O(n)$', where 'n' represents the sample size. The time complexity involved in outlier detection is mathematically expressed as given below.

$$TC = \sum_{i=1}^{n} Samples_i * Time\left(CSM_p(p,q) + CSM_q(p,q)\right) \quad (17)$$

From the above equation (17), the time complexity 'TC' is measured based on the stock market samples involved in simulation process '$Samples_i$' and the actual time consumed '$Time\left(CSM_p(p,q) + CSM_q(p,q)\right)$'. It is measured in terms of milliseconds (ms).

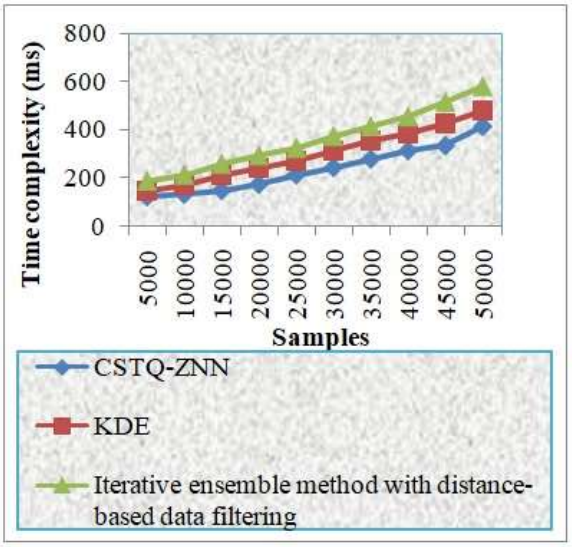

Figure. 4 Graphical representation of time complexity

Figure. 4 presents the time complexity by varying the samples obtained from NIFT-50 Stock Market Dataset as input. The results reveal clear superiority of the proposed CSTQ-ZNN compared to KDE Angiulli et al. [3] and Iterative ensemble method with distance-based data filtering Chakraborty et al.[7]. Indeed, the time complexity of the CSTQ-ZNN method does not go below 125ms and exceed 415.55ms, where KDE KDE Angiulli et al. [3] does not exceed 480ms, and Iterative ensemble method with distance-based data filtering Chakraborty et al.[7] does not exceed 580.15ms.

From these results it is inferred that though an increasing trend with respect to time complexity is observed when increased with the samples, however was found to be comparatively better using CSTQ-ZNN method than Angiulli et al. [3] and Chakraborty et al. [7]. The reason behind the improvement was owing to the application of Angles' Differences of Correlation of Quartile as a pre-processing step. Here, the redundant data are removed therefore obtaining processed data for further processing. This in turn minimizes the time complexity involved using CSTQ-ZNN method by 22% compared to Angiulli et al. [3] and 35% compared to Chakraborty et al. [7].

Case Scenario 2 : Error Tolerance

The part of the error tolerance remains in enabling detection and discarding of outliers. On the basis of the dataset there remains certain amount of apparent limits for error that should be utilized when obtainable. Hence, the value of error tolerance should be selected in such a manner as that the subset comprises of a large amount of non-outliers.

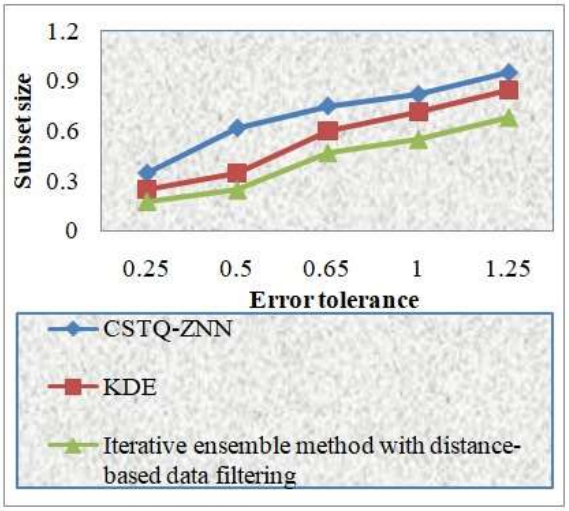

Figure.5 Graphical representation of error tolerance

Figure. 5 presents the error tolerance rate by varying the percentage of the subset size. The results reveal again clear superiority of the proposed CSTQ-ZNN compared to KDE Angiulli et al. [3] and Iterative ensemble method with distance-based data filtering Chakraborty et al. [7]. Indeed, the error tolerance of the CSTQ-ZNN method does not go below 0.35 and exceed 0.95, where KDE Angiulli et al. [3] does not exceed 0.85, and SVM does not exceed 0.68. Here, the subset size is found to be larger using CSTQ-ZNN method upon comparison with Angiulli et al. [3] and Chakraborty et al. [7].

From this result it is inferred that the value of error tolerance directly influences the size of the subset that fits the resulting model. Ideally, the error tolerance should be designed in such a manner that includes large amount of non-outliers. Also from the simulation results error tolerance was found to be better using CSTQ-ZNN method upon comparison with Angiulli et al. [3] and Chakraborty et al. [7]. The reason behind the improvement was owing to the application of Centroid Stabilized Fuzzy Tukey Quartile-based Clustering algorithm. By applying this algorithm, the raw data was stored in kernel matrix. Followed by which noise reduction was made by applying Angles' Differences of the Correlation of Quartile function. Furthermore, instead of employing an arbitrary centroid, Centroid Stabilization function was employed. With this the error tolerance using CSTQ-ZNN method was found to be better by 34% compared to Angiulli et al. [3] and 78% compared to Chakraborty et al. [7] respectively.

Case Scenario 3 : True Negative Rate

True negative rate is measured to estimate the efficiency of the outlier detection method. This is mathematically formulated as given below.

$$TNR = \frac{TN}{TN+FP} \quad (18)$$

From the above equation (18), the true negative rate 'TNR' is evaluated based on the true negative rate 'TN' and the false positive rate 'FP' respectively.

In terms of true negative rate, the results are shown in fig. 6 the results validate the obtained ones in the previous experiments, where superiority of the CSTQ-ZNN method is validated on samples ranging between 5000 and 50000. Responsible for these stimulating results is the effective decomposition approach that permits the entire data points to split divided into homogeneous clusters. This in turn permits to better train the different models of generated clusters.

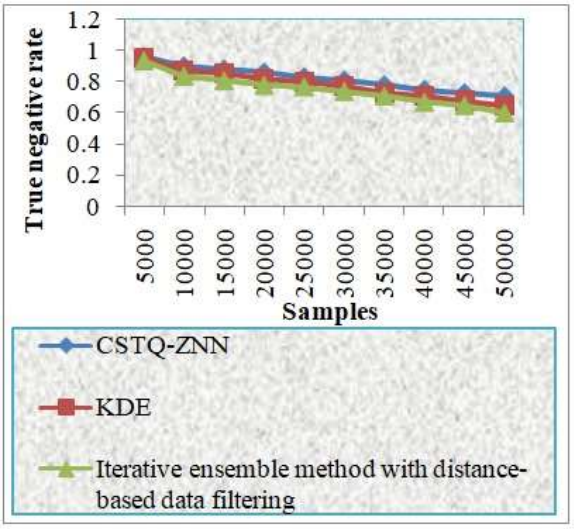

Figure.6 Graphical representation of true negative rate

On the other hand, KDE Angiulli et al. [3] and Iterative ensemble method with distance-based data filtering-based solutions Chakraborty et al. [7], that process all the data points at once, do not ensure this process. Moreover by utilizing the maxout as activation function reconstruct the data point in the corresponding cluster with multi-layer neurons (i.e., data points). This in turn improves the true negative rate using CSTQ-ZNN method by 5% compared to Angiulli et al. [3] and 10% compared to Chakraborty et al. [7].

Case Scenario 4 : Outlier Detection Rate

The outlier detection rate indicates how many outlying data points are exactly detected by an outlier method. The outlier detection rate is measured as given below.

$$ODR = \sum_{i=1}^{n} \frac{OD_i}{n} * 100 \quad (19)$$

From the above equation (19), the outlier detection rate 'ODR' is measured based on the percentage ratio of number of outlier detected 'OD_i' and the total number of outliers in data point 'n'. It is measured in terms of

percentage (%).

Finally, figure. 7 given below illustrates the outlier detection rate with respect to 50000 samples. A decreasing trend is found in all the three methods with an increase in the samples. However, out of 5000 samples, 45 data points detected as outlier, 40 data points were detected using CSTQ-ZNN, 37 data points were detected using KDE and 34 data points were detected using Iterative ensemble method with distance-based data filtering, then, the outlier detection rate using the three methods were observed to be 88.88% using CSTQ-ZNN method, 82.22% using Angiulli et al. [3] and 75.55% using Chakraborty et al.[7] respectively.

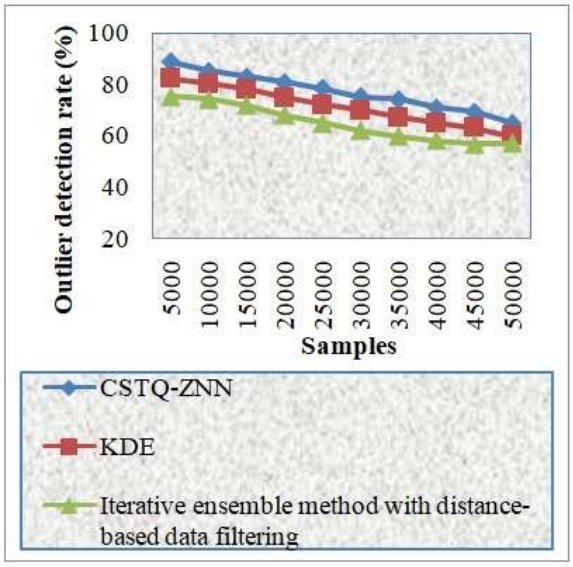

Figure. 7 Graphical representation of outlier detection rate

From this result, the outlier detection rate was observed to be better using CSTQ-ZNN method upon comparison to Angiulli et al. [3] and Chakraborty et al. [7]. The improvement in outlier detection rate using CSTQ-ZNN method was owing to the application of Deep Z-Curve Neural Network-based Outlier Detection algorithm. By applying this algorithm, a deep neural network using auto encoder and decoder was utilized with the input forming data points from processed clusters and the actual outlier detection results in the output. Moreover, the affinities between new representations were also fine tuned based on the discriminative information via maxout function. This in turn improved the outlier detection rate using CSTQ-ZNN method by 8% compared to Angiulli et al.[3] and 19% compared to Chakraborty et al. [7].

sumption and increasing the accuracy.

7 Conclusion

In this paper, we have investigated the challenge about how to detect the outliers in an efficient manner using fuzzy clustering and simultaneously achieve high detection rate with improved error tolerance and time complexity. We have proposed Centroid Stabilized Tukey Quartile and Z-curve Neural Network (CSTQ-ZNN) for outlier detection by first pre-processing the data points, which makes it possible to access only error minimized data points in a data asset rather than access all data points. To further enhance the performance of time complexity and error tolerance, Centroid Stabilized Fuzzy Tukey Quartile-based Clustering model is employed that obtains time efficient cluster results. Next, with the processed cluster results, Deep Z-Curve Neural Network-based Outlier Detection algorithm is applied that in turn improves outlier detection rate considerably. Evaluation results using NIFT-50 Stock Market Dataset have demonstrated that with our method, the efficiency of outlier detection can be significantly improved in terms of time complexity, error tolerance, true negative rate and outlier detection rate over existing methods.

References

[1] Afzal, S., Afzal, A., Amin, M., Saleem, S., Ali, N., Sajid, M. A Novel Approach for Outlier Detection in Multivariate Data. *Mathematical Problems in Engineering*, 1-12, 2022.

[2] Alharbi, F., Hindi, K. E., Ahmadi, S. A., and Alsalamn, H. Convolutional Neural Network-Based Discriminator for Outlier Detection. *Computational Intelligence and Neuroscience*, 1-13, 2021.

[3] Angiulli, F., Fassetti, F., Palopoli, L., and Serrao, C. A density estimation approach for detecting and explaining exceptional values in categorical data. *Applied Intelligence*, 1-23, 2022.

[4] Bertsimas, D., Orfanoudaki, A., and Wiberg, H. Interpretable clustering: an optimization approach. *Machine Learning*, 110:89–138, 2021.

[5] Bushra, N., and Rohli, R. V. Spatiotemporal Trends and Variability in the Centroid of the Northern Hemisphere's Circumpolar Vortex. *Earth and Space Science*, 1-16, 2021.

[6] Cai, S., Chen, J., Yin, B., Sun, R., Zhang, C., Chen, H., Chen, J., and Lin, M. An Efficient Outlier Detection Approach for Streaming Sensor Data Based on Neighbor Difference and Clustering. *Security and Communication Networks*, 2022:1-14, 2022.

[7] Chakraborty, B,. Chaterjee, A., Malakar, S., and

Sarkar, R. An iterative approach to unsupervised outlier detection using ensemble method and distance-based data filtering. *Complex & Intelligent Systems*, 8 (2): 3215–3230, 2022.

[8] Choi, K., Yi, J., Park, C., and Yoon, S. Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines. *IEEE Access*, 9: 120043 – 120065, 2021.

[9] Diers, J., and Pigorsch, C. Self-supervised learning for outlier detection. *Wiley*, 1-10, 2020.

[10] Dutta, J. K., Banerjee, B., and Reddy, C. K. RODS: Rarity based Outlier Detection in a Sparse Coding Framework. *IEEE Transactions on Knowledge and Engineering*, 28(2): 483 – 495, 2016.

[11] Feng, Y., Cai, W., Yue, H., Xu, J., Lin, Y., Chen, J., and Hu, Z. An improved X-means and isolation forest based methodology for network traffic anomaly detection. *PLOS ONE*, 17 (1): 1-18, 2022.

[12] Hooshmand, M. K., and Hosahalli, D. Network anomaly detection using deep learning techniques. *CAAI Transactions on Intelligence Technology*, 2021.

[13] Li, G., and Jung, J. J. Dynamic graph embedding for outlier detection on multiple meteorological time series. *PLOS ONE*, 1-14, 2021.

[14] Moens, S., Cule, B., and Goethals, B. RASCL: a randomised approach to subspace clusters. *International Journal of Data Science and Analytics*, 14:243–259, 2022.

[15] Nordahl, C., Boeva, V., Grahn, H., and Netz, M. P. (2021) EvolveCluster: an evolutionary clustering algorithm for streaming data. *Evolving Systems*, 13 (5): 603–623, 2021.

[16] Salem, S. B., Naouali, S., and Chtourou, Z. A rough set based algorithm for updating the modes in categorical clustering. *International Journal of Machine Learning and Cybernetics*, 12 (5): 2069–2090, 2021.

[17] Vasconcelos, I., Vasconcelos, R. O., Olivieri, B., Roriz, M., Endler, M., and Junior, M. C. Smartphone-based outlier detection: a complex event processing approach for driving behavior detection. *Journal of Internet Services and Applications*, 8(1): 1-30, 2017.

[18] Venkateswarlu, Baskar, Y. K., Wongchai, A., Shankar, V. G., Carranza, C. P. M., Gonzáles, J. L. A., and Dharan. A. R. M. An Efficient Outlier Detection with Deep Learning-Based Financial Crisis Prediction Model in Big Data Environment. *Computational Intelligence and Neuroscience*, 1-10, 2022.

[19] Wei, Y., Jang-Jaccard, J., Sabrina, F., and Alavizadeh, H. Large-Scale Outlier Detection for Low-Cost PM10 Sensors. *IEEE Access*, 8: 229033 – 229042, 2020.

[20] Yanxia, L., Wenjie, L., Yue, W., Siqi, S., and Cuirong, W. RMHSForest: Relative Mass and Half-Space Tree Based Forest for Anomaly Detection. *Chinese Journal of Electronics*, 29 (6): 1-9, 2020.

[21] Yoshihara, K., and Takahashi, K. A simple method for unsupervised anomaly detection: An application to Web time series data. *PLOS ONE*, 17(1): 1-25, 2022.

[22] Yuan, M., Zobel, J., and Lin, P. Measurement of clustering effectiveness for document collections. *Information Retrieval Journal*, 1-30, 2022.

[23] Zhou, W., Wang, L., Han, X., Parmar, M., and Li, M. A novel density deviation multi-peaks automatic clustering algorithm. *Complex & Intelligent Systems*, 1-35, 2022.