Heliyon 9 (2023) e17530

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

An enhanced self-learning-based clustering scheme for real-time traffic data distribution in wireless networks

Arpit Jain [a], Tushar Mehrotra [b], Ankur Sisodia [c], Swati Vishnoi [d], Sachin Upadhyay [e], Ashok Kumar [a], Chaman Verma [f,*], Zoltán Illés [f]

[a] Department of Computer Science & Engineering, Koneru Lakshmaiah Education Foundation (KLEF), Greenfield, Vaddeswaram, Guntur, Andhra Pradesh, 522302, India
[b] Department of Computer Science & Engineering Sharda School of Engineering and Technology, Sharda University, Greater Noida, India
[c] Department of Computer Engineering and Applications GLA University, Mathura, India
[d] Department of Computer Science and Engineering, Sanskriti University, Mathura, India
[e] Department of Computer Science and Engineering, GLA University Mathura, India
[f] Department of Media and Educational Informatics, Faculty of Informatics, Eötvös Loránd University, 1053 Budapest, Hungary

ARTICLE INFO

Keywords:
Wireless networks
Network traffic
Computation time
Clustering accuracy
Overhead

ABSTRACT

The process of examining the data flow over the internet to identify abnormalities in wireless network performance is known as network traffic analysis. When analyzing network traffic data, traffic classification becomes an important task. The traffic data classification is used to determine whether data in network traffic is in real-time or not. This analysis controls network traffic data in a network and allows for efficient network performance improvement. Real-time and non-real-time data are effectively classified from the given input data set using data mining clustering and classification algorithms. The proposed work focuses on the performance of traffic data classification with high clustering accuracy and low Classification Time (CT). This research work is carried out to fill the gap in the existing network traffic classification algorithms. However, the traffic data classification remained unaddressed for performing the network traffic analysis effectively. Then, we proposed an Enhanced Self-Learning-based Clustering Scheme (ESLCS) using an enhanced unsupervised algorithm and adaptive seeding approach to improve the classification accuracy while performing the real-time traffic data distribution in wireless networks. Test-bed results demonstrate that the proposed model enhances the clustering accuracy and True Positive Rate (TPR) effectively as well as reduces the CT time and Communication Overhead (CO) substantially to compare with the peer-existing routing techniques.

1. Introduction

The technique of interrupting and analyzing the messages in a network communication system to identify data trends is known as network traffic analysis [1]. Even without decrypting the messages, it is used in the context of military intelligence, counterintelligence, or pattern-of-life analysis [2]. The more the number of observed messages the more traffic needs to be monitored. Moreover, the invaders/intruders also utilize the network traffic analysis for studying the network traffic patterns and to detect the loopholes

* Corresponding author.
E-mail addresses: dr.jainarpit@gmail.com (A. Jain), tusharmehrotra9@gmail.com (T. Mehrotra), ankur22887@gmail.com (A. Sisodia), swativishnoi1@gmail.com (S. Vishnoi), sachin.upadhyay@gla.ac.in (S. Upadhyay), ashok_gangwar@rediffmail.com (A. Kumar), chaman@inf.elte.hu (C. Verma), illes@inf.elte.hu (Z. Illés).

https://doi.org/10.1016/j.heliyon.2023.e17530
Received 21 December 2022; Received in revised form 14 June 2023; Accepted 20 June 2023

Available online 28 June 2023

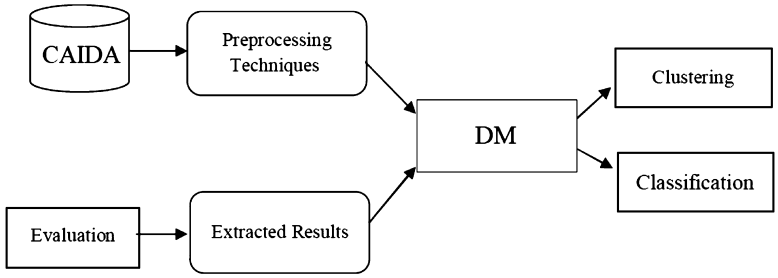

Fig. 1. Framework of network traffic analysis.

to misuse the sensitive data. Network traffic analysis is also employed in determining the type of data flow using physical and mechanical algorithms [3].

The first step in network traffic analysis is network traffic classification, which classifies or identifies various applications and protocols that are present in a network. Network classification is a key component of network Intrusion Detection Systems (IDS) [4], which use categorized network traffic to carry out operations like monitoring, discovering, controlling, and optimizing. Classifying network traffic is a crucial task that helps Internet Service Providers (ISPs) [5] identify the different types of application flows in a network. ISPs or network operators can control a network's overall performance with the aid of network traffic classification [6].

1.1. Network traffic analysis

When there are many networks and network applications in a network infrastructure, it is one of the most important challenges to quantifying the performance of the network. Network traffic analysis is a significant task that provides efficient troubleshooting algorithms to resolve many issues in the network and facilitates network services for an unlimited period of time [7]. It is the process of observing and investigating the network traffic to recognize the traffic flows in a network environment. Network traffic analysis is a proactive approach for guaranteeing secure, consistent, and qualitative network communication services. The network is examined at three different levels namely, packet level, flow level, and network level for security administration [8]. Different algorithms are being employed in network traffic analysis namely port-based, payload-based, and Machine Learning (ML) based algorithms [9]. Fig. 1, shows the framework of the general network traffic analysis process.

As shown in Fig. 1, network traffic analysis consists of preprocessing algorithms followed by Data Mining (DM) algorithms for analyzing and evaluating the patterns of the network data. A detailed description of the network traffic analysis is as follows:

- **Datasets:** Testing and assessing are the essential tasks of network traffic analysis for detecting the flow of traffic data. A standard Center for Applied Internet Data Analysis (CAIDA) [10] dataset is employed for computing the network traffic analysis for various network applications [11].
- **Preprocessing:** Preprocessing is applied as the second phase for changing the real-world data into a reasonable format. The real-world data is frequently incomplete and consists of noisy characteristics in a particular situation. Moreover, the data evaluated from the real world using DM algorithms are imperfect, and incompatible and also it includes errors and outlier values [12]. Therefore the preprocessing algorithms are mandatory to evaluate the DM algorithms which enhance the data quality and accuracy with better efficiency. Since there are several patterns and formats, the preprocessing methods are essential to recognize those patterns and formats during the network traffic analysis [13].
- **DM:** DM is employed for knowledge discovery and it plays a vital role in analyzing network traffic [14]. Two important procedures for DM algorithms are,
  - **Clustering algorithm:** Clustering is the process of categorizing the given dataset into clusters or groups for their characteristics. Clustering helps in dividing the data into a group of related objects and every group is labeled as a cluster. Each cluster consists of similar members and members of one cluster are diverse from the other one [15,16]. Clustering methods are generally employed to generate groups in network traffic data during network traffic analysis.
    Online Efficient Incremental Clustering (OEIC) [17] algorithm to cluster the network traffics. The huge networks required a predefined number of clusters (κ) and threshold values to cluster the network data. Most of the time these values produce wrong outcomes. The online Efficient Incremental Clustering algorithm was represented as an optimum algorithm for online clustering since it does not require more clusters (κ) and threshold values [18]. Therefore the increased level of accuracy with reduced time was attained in traffic clustering by employing the Online Efficient Incremental Clustering algorithm [19].
  - **Classification algorithm:** Classification is a function of data analysis that considers the search instance of a dataset and allocates it to a specific class. A classification-based network traffic analysis aids in categorizing the total traffic into normal and malicious. The main objective of the classification algorithm is to reduce the number of false positives and false negatives (i.e., target class incorrectly identified from traffic data). It also assists in predicting the target class accurately from the traffic data thus increasing the true negative rate (i.e., target class correctly identified from traffic data). The classification algorithm can recognize the network applications as low, medium, or high risk during the traffic analysis process [20].

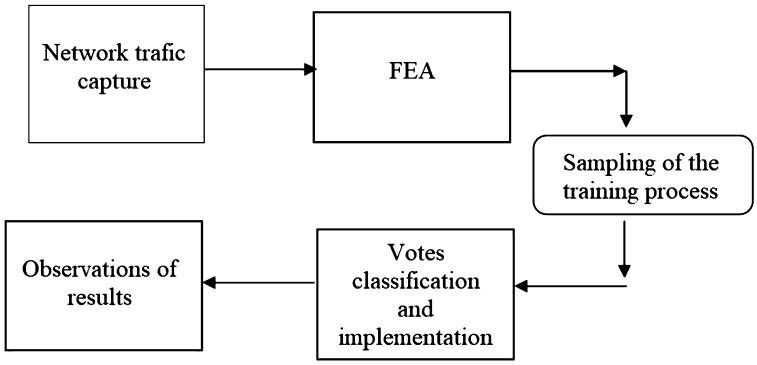

Fig. 2. Process of network traffic classification.

- **Evaluation:** DM algorithm practices a number of metrics for analyzing the traffic data. Once the DM processes are performed, the extracted result is passed through several other performance metrics like accuracy, TPR, and CO for further analysis [21].

1.2. Network traffic classification

Network traffic classification is a programmed process that classifies the network traffic into traffic classes depending on the port numbers or protocols. Each consequential traffic class is applied in a different way to distinguish the services provided to the user. The data packets are categorized and processed differently by the network scheduler. Since the traffic flow is classified using a specific protocol, a determined policy can be enabled for data and other flows. These policies ensure a specific quality offered to the data and other flows with an optimum delivery effort [22]. Network classification algorithms are applied to the point at which the traffic enters the network with granularity. This algorithm provides the traffic management mechanisms to partition the traffic into specific flows and queues and also each partition is shaped differently. Fig. 2, illustrates the process of network traffic classification.

The network traffic classification process consists of five steps as follows,

- **Network traffic capture:** Network Traffic Capturing is the first and most significant step in the classification process which executes the data collection from the network traffic. As a result, the real-time network traffic is captured with the help of various tools in an effective manner [23].
- **Feature Selection (FS) and extraction:** Once the network traffic data is captured, the FS and the extraction step are processed. During the FS process, the features such as packet duration, packet length, inter-arrival packet time, protocol, etc., are mined from the captured data. Then the extracted features are engaged in guiding the ML classifier [24].
- **Sampling and training process:** The datasets are sampled either for supervised or unsupervised learning algorithms for the applications in the network [25].
- **Implementation of ML algorithms:** The implementation step incorporates the procedures for applying the supervised, unsupervised and semi-supervised ML algorithms or classifiers or algorithms used in network traffic classification [26].
- **Results and observations:** The implementation of ML algorithms provides complete detail about the information accuracy, training time, or CT and TPR for further analysis [27].

  The Transparency, Consent, and Control (TCC) [28] information algorithm for traffic classification. It is a non-parametric approach to produce better performance in classification. TCC was utilized in the automatic recognition of unidentified network applications from the confined network traffic and a semi-supervised DM approach for handling the network packets. Additionally, three other classification algorithms namely Average Nearest Neighbor (AVG-NN) [29], Minimum Nearest Neighbor (MINNN), and Mean Variant Nearest Neighbor (MVT-NN) [30] were employed for inculcating the correlation information into the class prediction to boost the classification performance of the TCC framework.

  The Self-Adaptive Network Traffic Classification system (SANTaClass) [31] evades the manual interference needed to extend the exact payload-based signatures for different network applications in the real-time traffic classification. SANTaClass integrated the automated signature generation algorithms with real-time traffic classifiers. Furthermore, the automatic learning application stuff makes this SANTaClass algorithm mingle with any network. The signature generation algorithm was employed for recognizing the invariant patterns and also managing the text-based and binary-based encrypted applications consistently. SANTaClass also made use of an incremental learning algorithm for adjusting the dynamic nature of the network traffic through the construction of signatures.

1.3. Clustering scheme for effective network traffic analysis

The clustering approach is developed for restricting the noise attributes and to enhance the accuracy of network traffic classification by opting for informative attributes and representative instances. In the clustering-classification approach, the traffic data is preprocessed and the unnecessary or irrelevant attributes from the global perspective are removed. Then, the k-means clustering

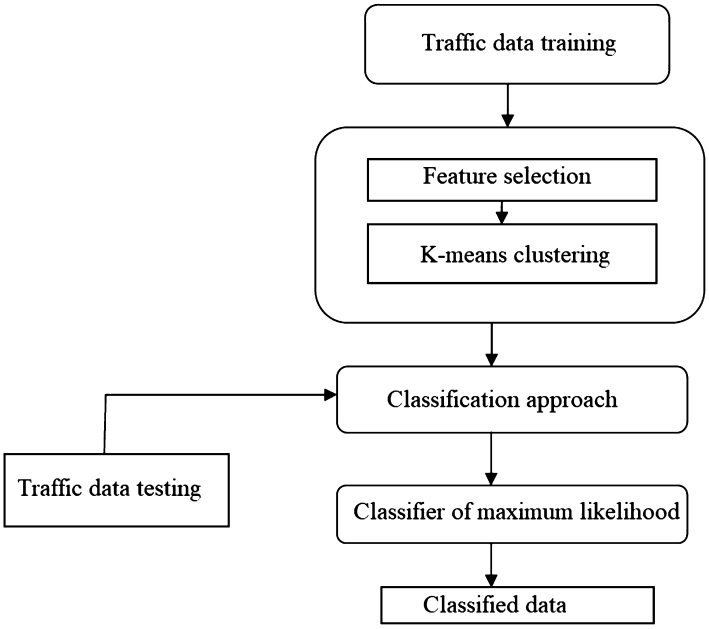

Fig. 3. Architecture of enhanced clustering scheme.

algorithm is applied to the training set for restricting the noisy instances and for the selection of a centroid for each cluster as the representative training instances. K-means clustering is a significant approach for some learning algorithms to minimize the amount of computation [32]. At last, a network classifier is applied to the representative training instances to determine the new traffic in real-time. Fig. 3 illustrates the architecture of the enhanced clustering scheme.

The enhanced clustering scheme consists of four steps as follows:

- The pre-processing of data for removing the inappropriate and unnecessary attributes from the original data from a global perspective.
- Identification of the most representative instances for enhancing the learning process efficiency and prediction accuracy by grouping the samples into a single class.
- Removal of the centroid of each cluster to perform as a representative instance of the application class.
- Finally, constructing the network traffic classification algorithm based on the network classifier performance.

1.4. Problem statement

A constrained clustering scheme referred to as the Set-Based Constrained K-Means (SBCK) algorithm was designed for enhancing the traffic CA. A constrained clustering scheme was implemented to observe the partial equivalence relationship among the IP flows to increase the clustering performance and speed up the convergence of the clustering process. After that, the decisions were implemented through the consideration of particular background knowledge and traffic statistics were examined. In the SBCK algorithm, a semi-supervised internet traffic clustering scheme was employed using the background information of traffic data to increase the clustering performance. This helped in clustering the internet traffic data more effectively. Subsequently, data and set-based limitations were fitted into Gaussian Mixture Model (GMM). The SBCK algorithm also improved the CA using the equal frequency binning method for feature discretization. However, the SBCK algorithm failed to cluster the real-time and non-real-time data.

The proposed ESLCS is described with a semi-automated internet flow traffic classifier for improving the accuracy of traffic data classification. In Adaptive Ranking Fuzzy-based Energy-efficient Opportunistic Routing (ARFOR), unsupervised clustering algorithms were employed for combining the data into clusters. After that, the filtering phase was developed to increase the clustering performance and coverage. An iterative clustering algorithm and self-seeding approach of the proposed algorithm reduced the number of clusters and increased the homogeneity. Even though the proposed algorithm is constructed with the help of an adaptive classification algorithm, iterative approach, layer-4 features, and iterative port filtering approaches, it failed to reduce the computational time during traffic classification.

The major contributions of the paper are listed below,

- Proposed algorithm analyzing the network traffic through the classification of patterns into real-time and non-real-time traffic data. Both unsupervised and supervised clustering and classification algorithms are used in the proposed algorithm.
- Proposed an effective traffic data classification to sustain the network traffic management.

- Proposed model is developed for improving the analysis of the network traffic data with reduced time consumption.

The following section depicts the general structure of the study method: Section 2 discusses the various peer-competing existing approaches and their limitations. Section 3 presents the proposed method and is tested with different scenarios. The result analysis of the proposed and existing protocols is presented in Section 4. Finally, Section 5 concludes the paper and highlights the future scope.

2. Literature review

Network traffic analysis is the process of tracking, assessing, and investigating the network traffic data for improving the performance and security of network processes. It is the process of utilizing physical and mechanical algorithms to evaluate the granular-level aspects and statistics in the network traffic. Network traffic analysis is employed in obtaining in-depth knowledge about the type of traffic or flow of data in a network. In general, the network traffic analysis is executed by observing the network or network bandwidth monitoring application. Malicious or suspicious packets in the traffic also can be recognized using the network traffic analysis.

Also, it involves examining the network traffic patterns and discovering the defects in sensitive data. Classification and clustering are the two important tasks for controlling network traffic. Supervised classification and unsupervised clustering are the common algorithms used in network traffic management. Even though there are many algorithms and approaches were established to enhance the performance of network traffic mining, there are many stumbling blocks in the network traffic analysis. Therefore, a few literature works are analyzed to enhance the traffic monitoring process for troubleshooting and solving the issues in the network services.

2.1. Literature on network traffic clustering schemes

Cheng et al. [33] developed a constrained clustering scheme for enhancing the accuracy of traffic clustering. A constrained clustering scheme was employed in constructing the decisions along with the background information on traffic statistics. A set of equivalent constraints was also incorporated into the constrained clustering scheme. Then, the monitored data with restrictions were used Gaussian mixture density and an approximate algorithm called SBCK algorithm to maximize the evaluation of algorithm parameters. In addition, a fundamental binning method was also utilized in recognizing the effect of unsupervised features in clustering. Therefore the constrained clustering scheme increases the CA but increased the time complexity as well.

Gao et al. [34] introduced the OEIC for clustering the network traffics. The networks with high rates required predefined numerous clusters (κ) and threshold values for clustering the network data. However, the outcomes from these predefined numerous clusters (κ) and threshold values produce erroneous results. Though the online efficient incremental clustering algorithm produced a higher level of accuracy with less time, it failed to address the issues of CO.

Whaiduzzaman et al. [35] presented an algorithm with two principles like Plackett-Luce (PL), Mixture Model (MM), and the iterative rank-tree algorithm for supervised clustering. The k random label rankings were altered and feature space was differentiated for decreasing the ranking loss after the re-calculation of the k rankings depending on the cluster assignments. MM-PL approach was a multi-prototype supervised clustering algorithm that depends on PL probabilistic ranking algorithm. MM-PL approach was employed in representing each cluster using a union of Voronoi cells with a set of prototypes and cluster central ranking was established by the allocation of each cluster with a set of PL label scores. But, PL Mixture Algorithm and the Iterative Rank Tree algorithm decreased the CA.

Zhang et al. [36] discussed a semi-supervised approach for enhancing internet traffic clustering. A GMM with set-based equivalence constraint and constrained EM algorithm was also involved in a semisupervised approach for clustering the traffic data. The quantization of feature flows in preprocessing step by a semi-supervised approach helped in increasing the traffic CA. Even though the traffic CA was increased, the CT was not reduced.

In network management, clustering is one of the significant tasks in traffic data depending on correlation analysis. Memon et al. [37] designed a similarity measure to evaluate the clusters of highly variable data for correlation. A similarity measure was applied in correlation-based clustering algorithms for time series analysis described by increased variability. Similarity measures solved the problems of conventional algorithms by improving the results of network and system measurements. Similarity measures also enhanced traffic clustering. But, still, it was unable to predict the traffic flows.

Lei et al. [38] developed congestion cluster identification for providing an essential amount of flexibility to various applications. A clustering algorithm was employed in minimizing the traffic network by avoiding repeated and congested clusters. The clustering method was applied to the city of Munich and the traffic conditions were predicted using floating car data. This helped in analyzing the congestion behavior of clusters. In addition, the negative effects of congestion were restricted by using this method to maintain the urban traffic but the traffic detection rate was less.

Ren et al. [39] analyzed the traditional clustering algorithms for a terrestrial social network to reduce the CO. The reduction of CO was performed by finding the message carriers which broadcast the messages nearer to the destination node. The bridge nodes between single-hop destination clusters were identified for guaranteeing the inter-cluster routing and to perform the alteration of existing schemes. This instigated a path toward a disjoint destination cluster. But, the CO issue was not addressed appropriately.

Chithaluru [40] elucidated the robust anomaly detection algorithm called Fuzzified Cuckoo-based Clustering Algorithm (F-CBCT). In F-CBCT two phases were involved namely training and detection. In the training phase, the Decision tree was employed along with the hybridization of Cuckoo Search Optimization and K-means clustering. Then, a multi-objective function depending on Mean Square Error (MSE) and Silhouette Index was utilized for calculating the two simultaneous distance functions such as the Classification

measure and Anomaly detection measure. When the system was trained, the detection phase was stimulated and a fuzzy decisive approach was utilized for identifying anomalies in the origin of input data and distance functions. F-CBCT increased the TPR in the detection of anomalies but CA was diminished.

2.2. Literature on network traffic classification schemes

Hewa et al. [41] presented a system called Principal Component Analysis of Packet Library (PCAPLib) to offer an efficient classification, extraction, and anonymized packet traces based on real network traffic. At first, the packet traces were extracted and classified by active trace collection through the influence of multiple detection devices. Then, deep packet anonymization was applied to secure the privacy of packet payload for many applications. PCAPLib system increased the TPR but failed to decrease the CT.

Chithaluru [13,42] introduced a semi-supervised classification method that depends on Particle Swarm Optimization (PSO). PSO was employed in identifying the centroids of classes and semi-supervised classification methods were used in attaining the merits of unlabeled occurrences. In semi-supervised PSO, restricted labeled samples and several unlabeled samples were employed in identifying the collection of prototypes for representing the patterns of entire data. Then, the unlabeled data was categorized with acquired prototypes in the principle of the “nearest neighborhood”. However, the time required for traffic classification was increased in the semi-supervised PSO method.

Wu et al. [43] introduced an algorithm called TCC information. The performance of classification was enhanced through the integration of correlated information into the classification process. TCC increased the traffic classification when compared with other conventional approaches. In addition, TCC was mostly used in the automatic recognition of unknown applications and employed in semi-supervised DM for dealing the network packets. Even though TCC minimized the CT, traffic prediction was not achieved effectively.

Chithaluru et al. [44] designed a Tailored Decision Tree Chain (T- DTC) model for better traffic classification. The imbalanced traffic distributions were utilized for improving the training and classification phases. Classification And Regression Task (CART) Decision Trees in the sequential chain were employed for avoiding the traffic applications and for removing the samples allocated to each application group. Then classifiers were applied for the applications that were not detected. However, T-DTC failed to reduce the CO.

Ullah et al. [45] presented an algorithm called Learning of Decomposable Models with Limited Cycle Size (LDMLCS) for traffic classification. LDMLCS addressed the issues of over-fitting and capturing the interaction between diverse features. Evaluation of marginal distributions in approximating model was performed and restriction of maximum cycle size in the graph was permitted to maintain the complexity of models. However, LDMLCS failed to minimize the CO.

Chithaluru et al. [46] presented a Traffic Classification method based on Expanded Vector (TCEV) for minimizing the number of packets used in classifying flows. In TCEV, seven types of relationships were developed for flows by considering the relationships between flows. The flow property was not required in TCEV and combined linear complexity of the flow number. TCEV achieved high performance with decreased number of processed packets and packet loss. But, TCEV increased the computational complexity.

2.3. Literature on network traffic analysis

Qiu et al. [47] introduced the Feature Extraction Approach (FEA) which provides robust user identification in the network environment. FEA employed only metadata of the traffic and generation of application-level user interactions. This allowed the additional and consistent identity with the richer discriminatory feature set. Then, the user-interaction-based feature extraction algorithm was examined for achieving increased recognition rates. This provided appropriate knowledge to the forensic investigators to remove, purify, and recognize the relevant network traffic effectively. Though the FEA enhanced user identification, the time consumed for traffic classification was not minimized.

Chithaluru et al. [48] discussed the FS algorithm for traffic classification with goodness, stability, and similarity. In the FS algorithm, the optimal features were integrated with outcomes of conventional FS algorithms for identifying reliable features. Then, the smallest sets of features were chosen for improving the data quality. FS enhanced the accuracy and improved the run-time performance of classifiers than the other traditional algorithms. However, the FS algorithm failed to enhance the traffic prediction rate.

Hao et al. [49] explained the feature extraction and selection approach. In the feature extraction and Selection approach, Wavelet Leaders Multi-fractal Formalism (WLMF) was employed for mining multifractal features from traffic for representing the traffic flows. Then, the obtained multi-fractal features were implemented with Principal Component Analysis (PCA) based FS method to eliminate the irrelevant and redundant features. Feature extraction and Selection approach increased the accuracy and was appropriate for real-time traffic classification due to its capability of categorizing traffic at the early stage of traffic transmission. But, time complexity was not reduced in the feature extraction and Selection approach.

2.4. Literature on classifiers for traffic data identification

Chithaluru et al. [50] described a modular, cascading traffic classification system called waterfall architecture for integrating the traffic classifiers. In waterfall architecture, cascade classification termed as a multiclassifier variant was employed for finding the IP transmissions. The problem of traffic classification was solved into smaller and independent modules to ease management. Then,

an optimization algorithm was employed in the automatic Selection of a set of best modules to maximize the performance in terms of CPU time, the number of errors, and the percentage of unrecognized flows. However, the classification accuracy was unable to improve at the required level.

Qiu et al. [51] introduced a malware behavioral classifier termed MultilAyer Graphs for MAlware detection (MAGMA). MAGMA depended on big data methodology determined by the real-world data acquired from traffic traces gathered in an operational network. Automatic extraction of patterns for particular input events was performed by MAGMA from a huge amount of events in the network. Then, a network connectivity graph was constructed which extracted the entire network behavior of the input or seed. The features were extracted from the connectivity graph and a supervised classifier was developed but minimized the TPR.

Chithaluru et al. [52] described a Self-Learning Intelligent Classifier (SLIC). A small number of training instances self learns were performed and implemented a classification model to attain high accuracy in categorizing non-static traffic flows. The classification model was constructed by a small set of labeled data and self-learns that were stimulated into chosen test instances. Then, self-learning was performed over the Naive Bayes which enabled the independent Selection of better candidates. In addition, SLIC was allowed to identify and categorize unknown application protocols. However, a method for decreasing CT remains unaddressed.

A major concern in different applications is network traffic, which is essential for the operation of the applications used in wireless sensor network [53]. It is described how to discover and maintain routes in sensor networks using routing protocols such as AODV, OLSR, and DSR with data compression [53]. Sisodia et al. [54] discussed various terms and routing protocols that were used in internet of things to provide better communication between sensor nodes. Shankhdhar et al. [55] discussed about the EEG signals for the operation of a drone and the analysis of human intent recognition.

2.5. Research gaps identified in the literature

The gaps identified in the literature are as follows,

- Most of the approaches failed to cluster the real-time and non-real-time data.
- Few approaches failed to reduce the computational time.
- The process of real-time and non-real-time traffic classification was complex in a few schemes.
- Few classification schemes are not considered real-time and non-realtime traffic data.

3. Proposed method

To overcome the existing limitations, the proposed CKM-DT with the CART algorithm was introduced for performing the network traffic analysis. The proposed CKM-DT algorithm classifies the patterns as real-time and non-real-time traffic data based on the network traffic conditions. The proposed Centroid-based K-Means cluster and Decision Tree (ECKM-DT) algorithm is performed in two phases namely clustering and classification. The proposed CKM-DT algorithm classifies the patterns as real-time and non-real-time traffic data based on the network traffic conditions. Initially, the unsupervised clustering is carried out using the Enhanced Centroid-based K-Means Unsupervised Clustering (ECKM-UC) method.

Through this ECKM-UC method, the κ initial (means) is created in a random manner using the data domain. The ECKM-UC method primarily identifies the number of clusters to be generated from the given input data points. After that, the centroid of each cluster is initialized. Following, the κ numbers of clusters are created with better results through the determination of the distance between the data points. Each cluster is formed by assigning the data points with minimum distance. Next, the centroid of the new cluster is recalculated and the data points are grouped under the new cluster. This clustering process is repeated until there is no data point to move into a new group. This leads to improvement of the CA. Later, the proposed ECKM-DT algorithm performs the classification process with the help of a decision tree generated by the CART supervised classification model. Through the decision tree with the CART supervised classification model, the decision tree is constructed to classify the network traffic data into real-time and non-realtime traffic data. In this manner, the classification of network traffic data is successfully performed with minimum time consumption.

In the network traffic analysis, cluster validation is an important entity that helps to enhance the CA and to ensure the effectiveness of traffic flow in the network environment. In a network, the features of traffic occurrences are identified through effective mining technology. In the proposed ECKM-DT method, unsupervised clustering is carried out with the introduction of the ECKM-UC method. Using this method, the number of given input data points is divided into k number of clusters with the nearest mean. The optimal results are discovered by introducing the cluster validation criterion. Thus, the proposed ECKM-DT algorithm uses the ECKM-UC method to cluster the traffic flow data points effectively from the dataset. Through this clustering process, the traffic flow data are identified and a similar group of traffic nodes is grouped. After the performance of unsupervised clustering on different input data points, the proposed ECKM- DT algorithm is applied for supervised classification. The classification of traffic data is performed by generating the decision tree using the CART classification model.

In the proposed ECKM-DT algorithm, the combination of unsupervised clustering and supervised classification is successfully carried out to perform efficient traffic data classification. Fig. 4 shows the overall structure of the centroid-based K-Means cluster and decision tree with the CART algorithm.

As shown in Fig. 4, the network traffic analysis is efficiently carried out with the implementation of the proposed ECKM-DT algorithm. Unsupervised clustering and supervised classification—the two key phases of the proposed algorithm—are used in network traffic analysis. To carry out the subsequent operations, initial fresh data (i.e., traffic data points) are taken from the CAIDA dataset.

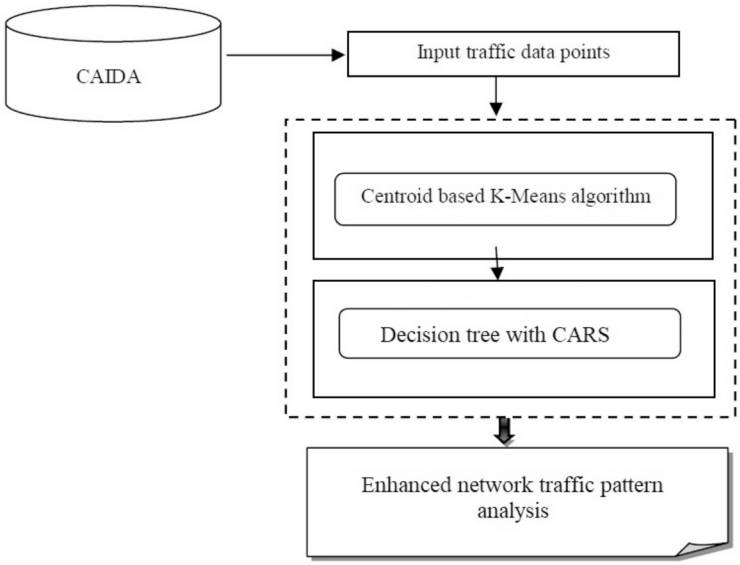

Fig. 4. Structure of ECKM-DT with CART algorithm.

The suggested algorithm manages the network infrastructure to actively track traffic on the chosen links. Numerous sequential workload traffic studies are included in the CAIDA traffic analysis. The collection of traffic data from a specific network node is used to calculate the workload. For instance, the data may be gathered by a traffic monitoring device and broadcast over a network link.

To carry out the clustering operation, the training dataset requires various parametric values. The evaluated parametric values are established on the validation set. The ECKM-UC method considers traffic data points from the given dataset as input to group similar traffic data points. The process of clustering is started by the identification of κ initial (means) from the given input dataset randomly. After that, the numbers of clusters with Cluster Heads (CHs) are generated with the nearest mean. For each cluster, the ECKM-UC method detects a centroid. The clustering operation is carried out with the determination of the distance between data points as well as the centroid of each cluster. According to the minimum distance, the clusters are generated by assigning each data point to a particular cluster. New clusters are generated and then the centroid of each new cluster is recalculated. This process ends when the dataset has no more data points to move. This procedure proves that the proposed ECKM-DT algorithm enhances the CA efficiently.

After that, the ECKM-DT algorithm introduces the effective decision tree with the CART supervised classification model to classify similar data points on the traffic data. The supervised approach works with the traffic-occurring data points in a network. The decision tree with a CART supervised classification model is used to classify real-time traffic data and non-realtime traffic data from the given input dataset. Moreover, this model also reduces the CT efficiency during the network traffic classification. The two essential phases involved in this model, make the network traffic data analysis more efficient.

3.1. ECKM-UC

In the proposed ECKM-DT algorithm, unsupervised clustering is the first process to cluster similar data points from the given dataset. The unsupervised approach manages the valuable frameworks in the knowledge acquisition process and detects the fundamental clusters on traffic network data points. Due to the involvement of the ECKM-UC method; the proposed ECKM-DT algorithm plays a vital role in computing the similarity of traffic flow in complex network traffic. The unsupervised DM approach is widely used to predict the target traffic rate.

The process of unsupervised clustering with varying parameters is depicted in Fig. 5. Using various input data points, the unsupervised clustering process creates the κ number of clusters. The clustering (grouping) process is carried out by the CH for each cluster group by the nearest mean. Each cluster group contains a black dot representing the cluster centroid. The ECKM-UC method is used in the proposed ECKM-DT algorithm to perform unsupervised clustering by dividing the η input data points into a specified number of partitions. The minimum distance between data points and the corresponding cluster centroid is used to cluster the input data points. After that, the data points are grouped into clusters that are comparable and have similar feature values. Assumed to be a positive integer number that represents the number of clusters in the ECKM-UC method, the κ parameter is assumed to be positive. The formation of three clusters, for instance, indicates that $\kappa = 3$.

Let us assume, η number of data points ξ_i, $i = 1,2,3, .., n$ grouped into κ number of clusters. The main purpose of unsupervised clustering is to construct the cluster for every input data point in the given dataset. The ECKM-DT algorithm introduces the ECKM-UC method to discover the positions α_i, $i = 1,2,3, .., n$ of the clusters by minimizing the distance between the centroids of the clusters. The clustering process is completed by achieving the lowest distance measure. The Euclidean distance function is mostly utilized to determine the distance between the data points. The Euclidean distance can be expressed mathematically as given in Eq. (1),

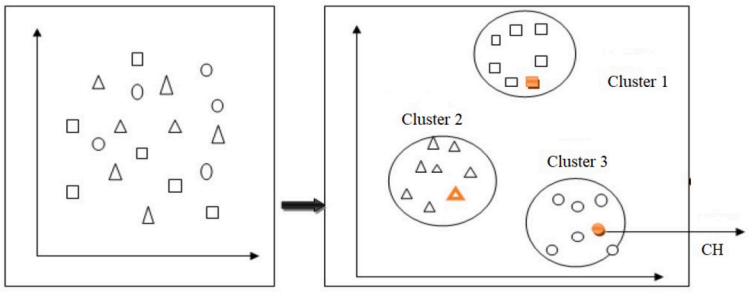

Fig. 5. Process of ECKM-UC.

$$D(p,q) = \sqrt{\sum_{i=1}^{q}(p_i - q_i)^2} \tag{1}$$

where (p,q) represents the two input vectors (two data points). All the data points equally contributed to the function value in the Euclidean distance function.

For each cluster, the cluster centroid is assumed by randomly separating all the data points into κ clusters through the Euclidean distance function. Every data point is considered a cluster centroid when data points are less than the number of clusters. Otherwise, the distances between the centroids are determined. Then, the data points with minimum distance are clustered. After placing all the data points in a specific cluster, when there is no data point to move towards a cluster the process gets finished. Otherwise, by recalculating the cluster and its centroid, the process gets repeated. Recalculation of centroid can be expressed mathematically as given in Eq. (2),

$$\beta = \frac{1}{\gamma_i}\sum_{i=1}^{n}\xi_i \tag{2}$$

where a set of cluster center β is indicated as $\beta = \{\beta_1, \beta_2, \beta_3, ..., \beta_n\}$. The centroid of the i^{th} cluster is represented by γ_i and the number of data points is represented by ξ_i.

The proposed ECKM-DT algorithm to cluster the input data points with higher CA. Initially, the ECKM-UC method defines the number of clusters κ to be generated with the consideration of data points from the given dataset. After grouping the data points, the clustered data points are classified by implementing the decision tree with the CART supervised classification model.

In the proposed ECKM-DT scheme, the supervised classification is the second phase. To enhance network traffic mining, huge numbers of network classifiers are tested with a random subset of features. But, the classification of real-time and non-real-time traffic is a difficult task during network analysis. Therefore, this research work introduces a decision tree with the CART supervised classification model to classify real-time and non-real-time traffic data successfully. The effective classification is carried out with the generation of a decision tree using a decision tree with the CART supervised classification model. This model is a well-organized ML algorithm to create exclusive decision trees.

In the ECKM-DT scheme, the CART supervised classification model is implemented to construct the decision trees from a training dataset by utilizing the concept of information entropy. Let us assume, τ_{S_i}, i = 1,2,3, .., n is the number of training dataset which is already classified. Then, the CART supervised classification model is initialized with a root node. The Decision tree with the CART supervised classification model identifies the attribute of the data at every node and classifies the samples into subsets. The splitting strategy is performed based on the normalized information gain. The attributes with the highest normalized information gain are identified as decisions in the decision tree algorithm. Then, the smaller sub-lists are returned by the CART algorithm. Fig. 6 illustrates the decision tree with the CART supervised classification model.

The supervised clustering approach with the objective of grouping the feature space into κ clusters. However, it failed to extract the feature to perform the classification process effectively. To overcome this limitation, the decision tree with the CART supervised classification model is implemented in the proposed ECKM-DT algorithm. As shown in Fig. 6, the decision tree is constructed with three attributes (P, Q, R). It shows that the attributes with higher information gain are chosen for making fine decisions. The classification rule is generated by each path from the root node to the leaf node in a decision tree. In addition, the recursive partition is also carried out in a particular time interval by the decision tree. Generally, the decision tree is limited to internal nodes, edges, and leaf nodes. Every internal node is labeled as a decision node which denotes the subset of attributes.

The number of classes is indicated as η and the number of instances is indicated as $\Gamma(P.i)$. Then, the entropy (ϵ) of attribute P is evaluated as given in Eq. (3),

$$\epsilon(P) = \sum_{i=1}^{n}\rho(A.i)log_2\rho(A.i) \tag{3}$$

Then, for a set of cases, the information gains ς is calculated. The confined cases' subset with different known values of the attribute from the tree is separated by the attribute (ς). The information gain is then assessed using Eq. (4),

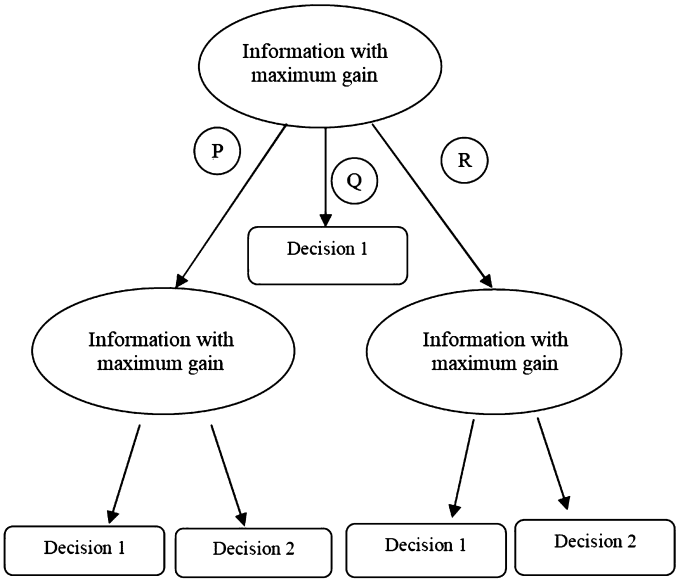

Fig. 6. Decision tree with CART supervised classification model.

$$\varsigma = \epsilon(P) - \sum_{i=1}^{n} \frac{|\varsigma_i|}{|\varsigma|} \epsilon(P_v) \tag{4}$$

whereas T_i represents a collection of P values in ς. In this case, ς_i denotes the subset of ς created by P and P_v denotes the attribute of P with the value v The gain ratio is then calculated using Eq. (5),

$$\%\varsigma(P,\varsigma) = \frac{Gain(P,\varsigma)}{spiltinfo(P,\varsigma)} \tag{5}$$

where splitinfo (P,ς) designates the data as a result of the division of T based on the value of As categorical attribute. The definition of $splitinfo(P,\varsigma)$ is then given in Eq. (6),

$$spiltinfo(P,\varsigma) = \sum_{i=1}^{n} \frac{|\varsigma_i|}{|\varsigma|} log_2 \frac{|\varsigma_i|}{|\varsigma|} \tag{6}$$

where the subsets of attributes are indicated as $|P_i|$ and $|P|$. Therefore, the information gain ratio is measured according to the decision tree and attributes with a huge number of different values. Then, for the determination of information gain, the attributes are selected which are more suitable. Therefore, the attributes are tested to verify whether they are nearest to the root of the tree. Then, the best attribute is identified and connected to the tree as given in Eq. (7),

$$P_{best} = argmax\{\%\varsigma(P,\varsigma)\} \tag{7}$$

where the attribute is selected with maximum information gain $argmax$. Then, the traffic analysis is carried out from the given input data by classifying them into real-time or non-real-time traffic data.

The ESLCS scheme to cluster the network traffic with the improvement in the robustness and accuracy beyond the huge stream of network traffic. But, the time remained unaddressed. Therefore, the proposed ECKM-DT algorithm uses a decision tree with CART supervised classification to reduce the CT and also minimize the CO. The process of decision tree with CART supervised classification algorithm to provide the accurate classification with minimal time.

In the proposed ESLCS scheme, clustering is the first process to group the input data points based on their similarity nature, which is performed by introducing the Enhanced Distribution based Expectation–Maximization Clustering (EDEMC) approach. The EM clustering is the iterative process that detects the Maximum Likelihood Estimate (MLE) of the data points from the dataset and groups them into related clusters. The MLE is the process of maximizing the likelihood by making observations for the data points.

According to the probability distribution function, the distribution-based Expectation Maximization (EM) clustering approach is carried out in two steps as Expectation step (ε) and the Maximization step (χ). The probability distribution function is determined as a number between '0' and '1'. Here, '1' indicates certainty which means that there is a chance (possibilities) for some event to have occurred and '0' indicates that there is no chance for an event to occur. In step ε, the probability function for the expectation of log-likelihood is determined by utilizing the current status of the data points. Then, step χ is carried out to re-estimate the distribution of data to maximize the expected log-likelihood.

The initial parameters (features or attributes) of distribution are randomly selected. Then, the two steps are carried out in the ESLCS approach to perform the traffic data clustering. The ε step is employed to identify the membership probability for each data

point by using the current parameters of the distributions. Later, based on the computed probabilities, the data points are relabeled. The similar parameters of data points are grouped into respective clusters. Then, the χ step re-estimates the parameters of the distributions. Then, the ESLCS approach generates the new clusters to maximize the likelihood of the data points.

Let us consider, η number of data points ξ_i, i = 1, 2, 3, ..., n grouped into γ number of clusters γ_i, i = 1, 2, 3, ..., n. Before performing the ϵ and χ steps in the distribution-based EM clustering, the set of vectors χ_η is extracted from the data points, which helps to cluster the same types of data points into respective clusters based on the features. The data points are indicated as $\xi_{i,\psi}$. Here, i denotes the data point and ψ denotes the feature of the data point. The set of features is randomly obtained as $\chi_{\eta,\psi}$ [0, 1]. Here, if the value is '1', then the feature ψ is to be used in classifying the data point ξ_i, or else ψ is ignored when it is '0'. The similar features of different data points are grouped into the same cluster. If the features helped with clustering, then it is considered unmasked, otherwise, it is considered masked (i.e. concealed). Then, the noise mean and the noise variance for feature ψ is calculated if the particular feature is masked i.e. $\chi_{\eta,\psi} = 0$. Therefore, the noise mean and the noise variance is determined as given in Eq. (8) and Eq. (9),

$$\alpha_\psi = \frac{1}{\omega A_f^{p\kappa}} \sum_{\eta:\chi_{\eta,\psi}=0} x_{i,\psi} \tag{8}$$

$$S_\psi^2 = \frac{1}{\omega A_f^{p\kappa}} \sum_{\eta:\chi_{\eta,\psi}=0} (x_{i,\psi} - \beta_I)^2 \tag{9}$$

where $\omega A_\psi^{p\kappa} = \{\eta : \chi_{\eta,\psi} = 0\}$, α_ψ denotes the noise mean and denotes the noise variance. Then, every data point x_n, with the virtual ensemble of $\tilde{x_n}$ is distributed as given in Eq. (10),

$$\tilde{x_\eta} = \begin{cases} x_{\eta,\psi} & \text{if } \chi_{\eta,\psi} \\ \omega(\alpha_\psi, S_\psi^2) & \text{if 1 - } \chi_{\eta,\psi} \end{cases} \tag{10}$$

As shown in Eq. (10), after the determination of the data points feature, the supporting quantities such as mean and variance are estimated before initializing the EM algorithm. The Mixture of Gaussians (MOG) model is initialized by soft assigning the parameters such as coefficient, mean, and covariance.

ϵ-step

In the ϵ step, the current values of parameters are used to evaluate the posterior probabilities for each data point. By using these probabilities, the expected clusters of all data points for each class are established. Therefore, the membership probability for all data points is determined as per Eq. (11),

$$A(x_i \in gamma_i) = p(\gamma_i|x_i)\frac{p(\gamma_i)p(\gamma_i|x_i)}{p(x_i)} \tag{11}$$

Where $A(x_i \in \gamma_i)$ represents the probability of data point forming clusters. Here, x_i denotes data points and γ_i denotes the clusters. So, by using the Eq. (11) all the data points are assigned into clusters by computing the membership probability of all data points.

χ-step

After performing the ϵ-step, the χ-step is carried out to maximize the likelihood of the data points. In χ-step, the distribution parameters such as mean, covariance, and mixing coefficient are re-estimated as given in Eq. (12), Eq. (13) and Eq. (14),

$$\alpha_c = \sum_{i=1}^{n} \frac{p(\gamma_i|x_i)}{n \times W_c} x_i \tag{12}$$

$$\sum \gamma = \sum_{i=1}^{n} \frac{p(\gamma_i|x_i)}{n \times W_c} (x_i - \alpha_c)^2 \tag{13}$$

$$W_c = \frac{1}{n} \sum_{i=1}^{n} p(\gamma_i|x_i) \tag{14}$$

Where α_c represents the mean, $\sum\gamma$ represents the covariance, and mixing coefficients is signified as W_c. After the re-estimation of parameters, the new clusters are generated. By forming the new clusters, the sum of the squared error or expected likelihood is maximized. The new clusters are formed by using maximum likelihood probability estimation m_c from the E-step as given in Eq. (15),

$$m_c = \frac{1}{n} \sum_{i=1}^{n} \frac{x_i A(x_i \in \gamma_i)}{\sum j A(x_i \in \gamma_i)} \tag{15}$$

From Eq. (15), the new clusters are formed to improve the CA. The re-estimation of distribution parameters helps to maximize the likelihood of the data points.

The maximum-likelihood estimation is employed for computing the value of one or more parameters for a given statistic which makes the known likelihood distribution a maximum. From the re-estimated values of parameters, the new clusters are formed which

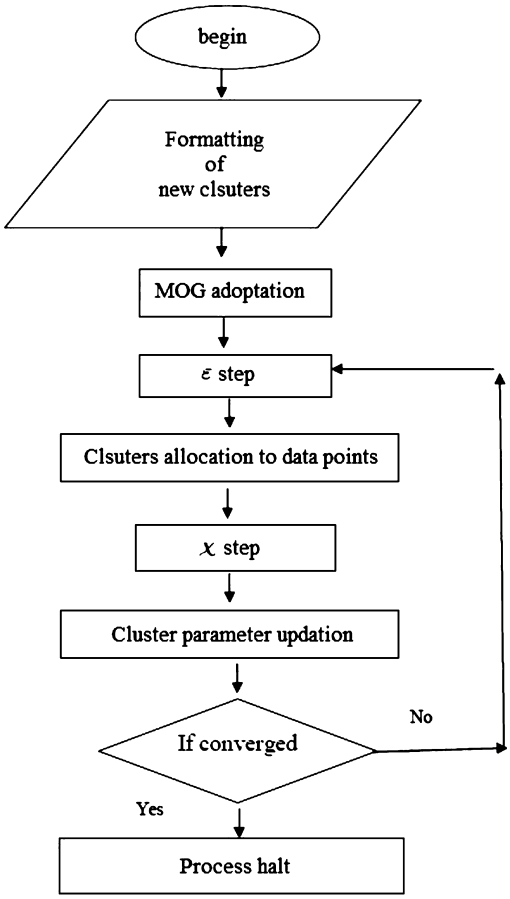

Fig. 7. Process of EDEMC approach.

leads to improving the CA. Then, the ε and χ steps are repeated until the parameters converge. Therefore, the proposed algorithm performs the EDEMC approach for clustering the data points with improved accuracy. Fig. 7 shows the process of EDEMC.

In Fig. 7, EM clustering is executed to cluster the input data points from the CAIDA dataset into a number of clusters with improved accuracy. At first, the MOG model is initialized. For each data point, the membership probabilities are computed. Based on the probability, the data points are clustered in the ε step. Then, the parameters are re-estimated in the χ-step. Accordingly, the clusters are updated. This process is continued until parameters converge. Finally, the data points are clustered with improved accuracy by using the EDEMC in the proposed ESLCS scheme.

Initially, the data points are taken from the CAIDA dataset. The distribution of parameters is selected randomly to cluster the data points. At first, the MOG is initialized by computing the parameters such as weight, means, and covariance. Then, the membership probability of data points is determined in step ε using Eq. (11). Then, the data points which have similar features are clustered into a number of clusters. Secondly, the distribution of parameters is re-estimated by using the maximum likelihood estimation through Eq. (15). Then, the new clusters are generated by maximizing the likelihood of the data points. As a result, the EDEMC approach improves the CA.

4. Results and discussions

In this section, the effectiveness of the proposed approach is proved by comparing the experimental results with the existing algorithms namely Multilayer Threshold Cluster-Based Energy-Efficient Low-Power and Lossy Networks (MTCEE-LLN) [13] and ARFOR [42]. According to the obtained performance values, the effectiveness of the proposed approach is proved.

4.1. Simulation environment

To analyze the network traffic data, an effective ESLCS scheme is experimented with using the CAIDA dataset. The proposed ESLCS scheme experimented using the Java platform, Intel(R) Core(TM) i5 processor with 1.62GHZ, 8 GB memory, 250 GB hard disk, and operating system is Windows 7. To determine the better performance of the proposed ESLCS algorithm, it is compared with the existing ARFOR and MTCEE-LLN algorithms based on the CA, CT, TPR, and CO.

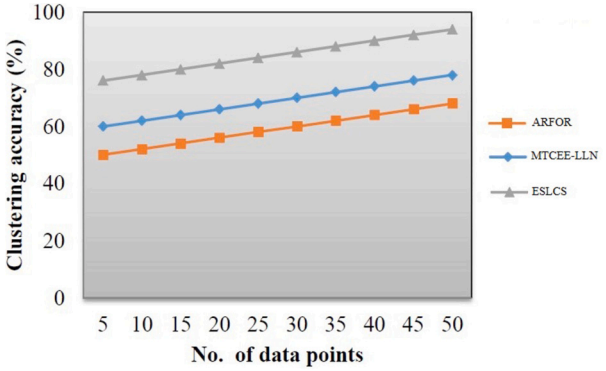

Fig. 8. Performance analysis of CA.

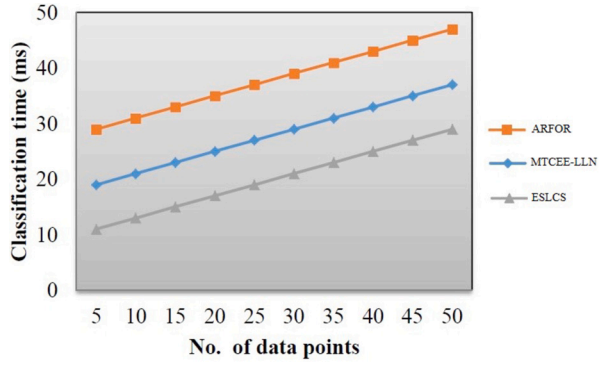

Fig. 9. Performance analysis of CT.

4.2. Evaluation parameters

Performance analysis CA The existing and proposed algorithms are compared based on the given input dataset. The comparison result of CA for the given input data point ranges from 5 to 50. The experimental result shows that the three algorithms enhance the CA for the given input dataset. However, the proposed approach enhances the CA more than the existing algorithms.

Fig. 8 provides the measurement of CA based on the different number of input data points. The simulation is conducted by comparing the proposed scheme with existing algorithms such as ARFOR and MTCEE-LLN algorithms. From Fig. 8, it is evidenced that the proposed algorithm provides the improved CA rather than the existing algorithms. This is because; the proposed approach effectively performs the clustering with the help of the EDEMC approach. Here, the data points from the input dataset are grouped based on similar features by executing the ε and χ steps. This, in turn, improves the accuracy of clustering the data points in the network traffic analysis. Moreover, the CA in the proposed approach is enhanced by 16% and 31% when compared to ARFOR and MTCEE-LLN algorithms respectively.

Performance analysis of CT

For the different numbers of input data points, the proposed algorithm is compared with the existing algorithms namely the ARFOR algorithm and MTCEE-LLN algorithm. The number of data points range is varied from 10 to 50, which has been taken as input to perform the experiments.

An experiment shows that the three algorithms are reducing the CT successfully. Comparatively the proposed algorithm consumes very less time to classify the traffic data than the other existing algorithms. Fig. 9 demonstrates the analysis of CT obtained from the different number of input data points. The simulation is conducted by comparing the proposed algorithm with existing algorithms namely ARFOR and MTCEE-LLN. Using the random forest classification algorithm, the proposed algorithm reduces the CT much better than the other existing algorithms. This is because the EDEMC-based ensemble classification algorithm is employed in the proposed algorithm to classify the traffic data after clustering the data with similar features. ECKM-DT and ECKM-UC are employed to partition the nodes into sub-nodes. The EDEMC-based ensemble classification algorithm generates the decision trees by ARFORing the features randomly which leads to forming the forest. After the construction of the decision trees with the number of nodes, the input data are classified effectively as real-time and non-real-time traffic data by considering the majority vote algorithm. As a result, the EDEMC-based ensemble classification algorithm consumes less time to classify the traffic data from the given input dataset. Moreover, the CT in the proposed algorithm is reduced by 26% and 36% when compared with ARFOR and MTCEE-LLN algorithms respectively.

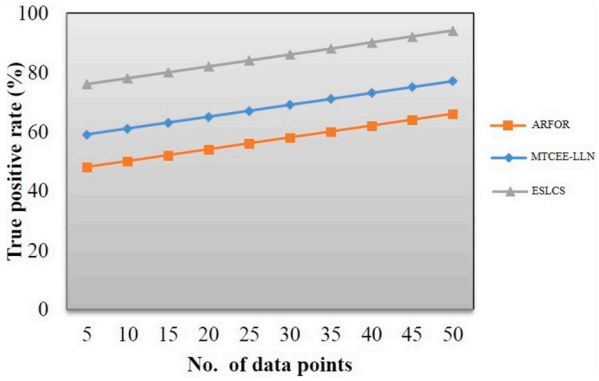

Fig. 10. Performance analysis of TPR.

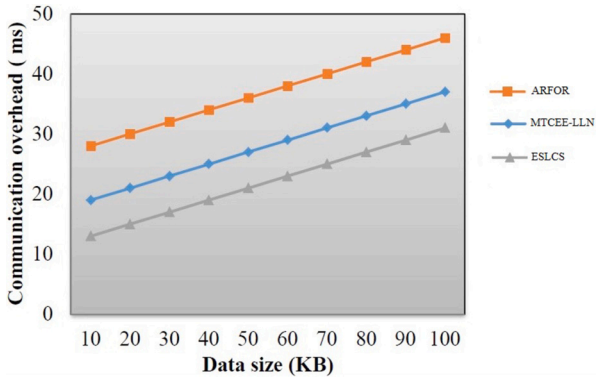

Fig. 11. Performance analysis of CO.

Performance analysis of TPR

The experimental result of TPR for the different number of input datasets is shown in Fig. 10. The number of data points range is varied from 5 to 50 which have been taken as input to perform the experiments. The performance analysis for TPR is carried out by comparing the proposed algorithm with existing algorithms such as ARFOR and the MTCEE-LLN algorithm.

The experimental results from Fig. 10 show that the three algorithms improve the TPR successfully and efficiently with the increased number of input data points. However, the proposed algorithm enhances the TPR more than the other existing algorithms. Fig. 10 shows the experimental results on TPR for the given set of input data points. Since the EDEMC-based classification technique, the TPR is enhanced efficiently in the proposed approach. The proposed approach produces improved results when compared to the existing algorithms because this technique will compensate for the disadvantages of the existing algorithm during traffic data classification. Moreover, experimental results show that the proposed algorithm enhanced by 19% and 41% when compared with ARFOR and MTCEE-LLN algorithms respectively.

Performance analysis of CO

Fig. 11 shows the comparison of CO for the given input data points. Here the proposed algorithm is compared with two existing algorithms namely ARFOR and MTCEE-LLN for the experimental purpose. The range of data size is varied from 10 to 100 which is taken as input to perform the tests.

Experimental analysis shows that the three algorithms are working in a positive direction to reduce the CO during the time of network traffic analysis. However, the proposed algorithm minimizes the CO much better than the other existing algorithms namely ARFOR and MTCEE-LLN. Fig. 11 provides the measurement of CO based on the various input data ranges. From the experimental outcomes, it is proved that the proposed algorithm reduces CO effectively. Due to this, the EDEMC is operated with a random forest-based ensemble classification algorithm which overcomes the shortcomings of the existing approach. This ensembling algorithm also classifies the network traffic into real-time and non-real-time traffic in an efficient manner. Moreover, the proposed algorithm improves the flow rate in the network with reduced CO. Additionally, the COin proposed algorithm is reduced by 18% and 36% when compared to ARFOR and MTCEE-LLN algorithms respectively. Table 1 shows the comparative analysis of the proposed method and peer-competing existing routing protocols.

Table 1
Comparative analysis of proposed with peer competing routing protocols.

Cluster routing Protocols	Network	Cluster formation	CH selection	Algorithm complexity	CH role	Process dynamic	Location awareness
Traditional	Homogeneous	Distributed	Random	$o(n^3)$	Relaying	Less	Required
MTCEE-LLN	Homogeneous	Centralized	Random	$o(n^3)$	Relaying	Less	Required
ARFOR	Homogeneous	Centralized	Random	$o(n^3)$	Relaying	Less	Required
Proposed	Heterogeneous	Centralized	Node parameters and network dynamics	$o(n^2)$	Aggregating	Highly dynamic	Not required

5. Conclusion and future scope

In this paper, we proposed the ESLCS algorithm to offer effective traffic data classification in the emerging network traffic management system. Two key processes such as clustering and classification are involved in the proposed algorithm. At first, the numbers of data points from the input dataset are grouped by implementing the maximum-likelihood mixture of ECKM with the help of the EDEMC approach. Through this maximum-likelihood estimation, the data points with similar features are clustered effectively to improve the CA. Secondly, the EDEMC-based ensemble classification algorithm is developed to classify the traffic data into real-time and non-real-time traffic data. After combining the decision trees, the attributes with the highest votes are identified as real-time traffic data and the other attributes are identified as non-real-time traffic data. So it proves that the classification is performed successfully with less CT. As a result, the flow rate is increased efficiency and the CO is minimized significantly. The experiment result exposes the proposed approach and improves the experimental outcomes, it is apparent that the proposed algorithm ensures better performance on the classification parameters such as CA, CT, TPR, and CO. Moreover, the experiment shows that TPR attains 19%, & 41% and CO is reduced by 18% & 36% when compared with the existing algorithms. The future direction of the proposed approach is concentrated on ensuring the secured traffic data analysis and also effective ensemble classifiers will be introduced to provide enhanced results in the classification process.

CRediT authorship contribution statement

Arpit Jain: Conceived and designed the experiments; Analyzed and interpreted the data; Wrote the paper
Ankur Sisodia: Conceived and designed the experiments; Wrote the paper
Tushar Mehrotra: Conceived and designed the experiments
Swati Vishnoi; Sachin Upadhyay: Performed the experiments; Contributed reagents, materials, analysis tools or data
Ashok Kumar: Analyzed and interpreted the data
Chaman Verma; Zoltán Illés: Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper

Declaration of competing interest

The authors declare that they have no competing interests.

Data availability

The data that has been used is confidential.

Acknowledgement

The work of Zoltán Illés and Chaman Verma was supported by the Department of Media and Educational Informatics, Faculty of Informatics, Budapest, Hungary.

References

[1] M. Bhende, A. Thakare, B. Pant, P. Singhal, S. Shinde, B.N. Dugbakie, Integrating multiclass light weighted BiLSTM model for classifying negative emotions, Comput. Intell. Neurosci. 2022 (2022).
[2] R. Tanwar, et al., Advanced Healthcare Systems: Empowering Physicians with IoT-Enabled Technologies, John Wiley & Sons, 2022.
[3] J. Agrawal, M. Gupta, H. Garg, Early stress detection and analysis using EEG signals in machine learning framework, IOP Conf. Ser., Mater. Sci. Eng. 1116 (1) (2021, April) 012134.
[4] P. Chithaluru, L. Jena, D. Singh, K.M. Ravi Teja, An adaptive fuzzy-based clustering model for healthcare wireless sensor networks, in: Ambient Intelligence in Health Care, 2023, pp. 1–10.

[5] T. Mehrotra, N. Shukla, T. Chaudhary, G.K. Rajput, M. Altuwairiqi, M. Asif Shah, Improved frame-wise segmentation of audio signals for smart hearing aid using particle swarm optimization-based clustering, Math. Probl. Eng. 2022 (2022).
[6] A. Shankhdhar, A. Mangla, A.K. Singh, A. Srivastava, Operating of a drone using human intent recognition and characteristics of an EEG signal, in: 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), IEEE, 2020, November, pp. 324–327.
[7] A. Yadav, et al., An enhanced feed-forward back propagation Levenberg–Marquardt algorithm for suspended sediment yield modeling, Water 14 (22) (2022) 37–44.
[8] D. Joshi, et al., An optimized open pit mine application for limestone quarry production scheduling to maximize net present value, Mathematics 10 (21) (2022) 41–57.
[9] P. Chithaluru, T. Stephan, M. Kumar, A. Nayyar, An enhanced energy-efficient fuzzy-based cognitive radio scheme for IoT, Neural Comput. Appl. 34 (21) (2022) 19193–19215.
[10] A. Jain, et al., Improved recurrent neural network schema for validating digital signatures in VANET, Mathematics 10 (20) (2022) 3895–4907.
[11] D. Joshi, et al., A novel large-scale stochastic pushback design merged with a minimum cut algorithm for open pit mine production scheduling, Systems 10 (5) (2022) 159–169.
[12] A. Yadav, et al., Suspended sediment yield forecasting with single and multi-objective optimization using hybrid artificial intelligence models, Mathematics 10 (22) (2022) 4263–4275.
[13] P. Chithaluru, et al., MTCEE-LLN: multilayer threshold cluster-based energy-efficient low-power and lossy networks for industrial Internet of things, IEEE Int. Things J. 9 (7) (2021) 4940–4948.
[14] S.K. Ramakuri, P. Chithaluru, S. Kumar, Eyeblink robot control using brain-computer interface for healthcare applications, Int. J. Mob. Devices Wearable Technol. Flex. Electron. 10 (2) (2019) 38–50.
[15] P. Chithaluru, R. Prakash, Organization security policies and their after effects, in: Information Security and Optimization, Chapman and Hall/CRC, 2020, pp. 43–60.
[16] P. Chithaluru, R. Tanwar, S. Kumar, Cyber-attacks and their impact on real life: what are real-life cyber-attacks, how do they affect real life and what should we do about them?, in: Information Security and Optimization, Chapman and Hall/CRC, 2020, pp. 61–77.
[17] P. Chithaluru, K. Singh, M.K. Sharma, Cryptocurrency and blockchain, in: Information Security and Optimization, Chapman and Hall/CRC, 2020, pp. 143–158.
[18] B.M.C. Chowdary, M.M. Sasank, P. Chithaluru, Design and development of novel flood detection system using IoT, Turk. J. Comput. Math. Educ. 11 (3) (2020) 1611–1620.
[19] P. Chithaluru, Energy efficient routing approach based on volunteer participation and adaptive ranking of forwarder nodes in WSNs, Doctoral dissertation, School of Computer Science, UPES, Dehradun, 2020.
[20] P. Chithaluru, R. Prakash, S. Srivastava, WSN structure based on SDN, in: Innovations in Software-Defined Networking and Network Functions Virtualization, IGI Global, 2018, pp. 240–253.
[21] T. Mehrotra, A review on attack in wireless and computer networking, Asian J. Multidimens. Res. 10 (10) (2021) 1457–1463.
[22] T. Mehrotra, G.K. Rajput, M. Verma, B. Lakhani, N. Singh, Email spam filtering technique from various perspectives using machine learning algorithms, in: Data Driven Approach Towards Disruptive Technologies, Springer, Singapore, 2021, pp. 423–432.
[23] A. Pandey, H. Jaiswal, A. Vij, T. Mehrotra, Case study on online fraud detection using machine learning, in: 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), IEEE, 2022, April, pp. 48–52.
[24] Y. Wu, Y. Ma, H.N. Dai, H. Wang, Deep learning for privacy preservation in autonomous moving platforms enhanced 5G heterogeneous networks, Comput. Netw. 185 (2021) 107743.
[25] C. Thapa, S. Camtepe, Precision health data: requirements, challenges and existing techniques for data security and privacy, Comput. Biol. Med. 129 (2021) 104130.
[26] W.A. Kassab, K.A. Darabkh, A–Z survey of Internet of things: architectures, protocols, applications, recent advances, future directions and recommendations, J. Netw. Comput. Appl. 163 (2020) 102663.
[27] D.A. Pustokhin, et al., Optimal deep learning approaches and healthcare big data analytics for mobile networks toward 5G, Comput. Electr. Eng. 95 (2021) 107376.
[28] M.H. Nasir, et al., Swarm intelligence inspired intrusion detection systems—a systematic literature review, Comput. Netw. (2022) 108708.
[29] M. Wu, et al., Unraveling the capabilities that enable digital transformation: a data-driven methodology and the case of artificial intelligence, Adv. Eng. Inform. 50 (2021) 101368.
[30] J. Curzon, A. Almehmadi, K. El-Khatib, A survey of privacy enhancing technologies for smart cities, Pervasive Mob. Comput. 55 (2019) 76–95.
[31] N. Kaaniche, M. Laurent, S. Belguith, Privacy enhancing technologies for solving the privacy-personalization paradox: taxonomy and survey, J. Netw. Comput. Appl. 171 (2020) 102807.
[32] M. Monshizadeh, et al., A deep density based and self-determining clustering approach to label unknown traffic, J. Netw. Comput. Appl. 207 (2022) 103513.
[33] X. Cheng, et al., Combating emerging financial risks in the big data era: a perspective review, Fundam. Res. 1 (5) (2021) 595–606.
[34] R.X. Gao, et al., Big data analytics for smart factories of the future, CIRP Ann. 69 (2) (2020) 668–692.
[35] M. Whaiduzzaman, et al., BFIM: performance measurement of a blockchain based hierarchical tree layered fog-IoT microservice architecture, IEEE Access 9 (2021) 106655–106674.
[36] Y. Zhang, et al., A privacy-aware PUFs-based multiserver authentication protocol in cloud-edge IoT systems using blockchain, IEEE Int. Things J. 8 (18) (2021) 13958–13974.
[37] R.A. Memon, et al., DualFog-IoT: additional fog layer for solving blockchain integration problem in Internet of things, IEEE Access 7 (2019) 169073–169093.
[38] K. Lei, et al., Groupchain: towards a scalable public blockchain in fog computing of IoT services computing, IEEE Trans. Serv. Comput. 13 (2) (2020) 252–262.
[39] J. Ren, et al., Task offloading strategy with emergency handling and blockchain security in SDN-empowered and fog-assisted healthcare IoT, Tsinghua Sci. Technol. 27 (4) (2021) 760–776.
[40] P. Chithaluru, et al., Energy-efficient blockchain implementation for cognitive wireless communication networks (CWCNs), Energy Rep. 7 (2021) 8277–8286.
[41] T. Hewa, et al., Fog computing and blockchain-based security service architecture for 5G industrial IoT-enabled cloud manufacturing, IEEE Trans. Ind. Inform. 18 (10) (2022) 7174–7185.
[42] P. Chithaluru, et al., An energy-efficient routing scheduling based on fuzzy ranking scheme for Internet of things, IEEE Int. Things J. 9 (10) (2021) 7251–7260.
[43] H. Wu, et al., EEDTO: an energy-efficient dynamic task offloading algorithm for blockchain-enabled IoT-edge-cloud orchestrated computing, IEEE Int. Things J. 8 (4) (2020) 2163–2176.
[44] P. Chithaluru, R. Tiwari, K. Kumar, Performance analysis of energy efficient opportunistic routing protocols in wireless sensor network, Int. J. Sens. Wirel. Commun. Control 11 (1) (2021) 24–41.
[45] Z. Ullah, et al., Towards blockchain-based secure storage and trusted data sharing scheme for IoT environment, IEEE Access 10 (2022) 36978–36994.
[46] P. Chithaluru, et al., ETH-LEACH: an energy enhanced threshold routing protocol for WSNs, Int. J. Commun. Syst. 34 (12) (2021) e4881.
[47] C. Qiu, et al., Networking integrated cloud–edge–end in IoT: a blockchain-assisted collective Q-learning approach, IEEE Int. Things J. 8 (16) (2020) 12694–12704.
[48] P. Chithaluru, R. Tiwari, K. Kumar, Arior: adaptive ranking based improved opportunistic routing in wireless sensor networks, Wirel. Pers. Commun. 116 (1) (2021) 153–176.
[49] X. Hao, et al., Stochastic Analysis of Double Blockchain Architecture in IoT Communication Networks, IEEE Internet of Things Journal, 2022.

[50] P. Chithaluru, et al., I-AREOR: an energy-balanced clustering protocol for implementing green IoT in smart cities, Sustain. Cities Soc. 61 (2020) 102254.
[51] C. Qiu, et al., Cloud computing assisted blockchain-enabled Internet of things, IEEE Trans. Cloud Comput. 10 (1) (2022).
[52] P. Chithaluru, R. Tiwari, K. Kumar, AREOR–adaptive ranking based energy efficient opportunistic routing scheme in wireless sensor network, Comput. Netw. 162 (2019) 106863.
[53] A. Sisodia, H.H. Swati, Incorporation of non-fictional applications in wireless sensor networks, Int. J. Innov. Technol. Explor. Eng. 9 (11) (2020).
[54] A. Sisodia, S. Kundu, Enrichment of performance of operation based routing protocols of WSN using data compression, in: 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART), IEEE, 2019, November, pp. 193–199.
[55] A. Sisodia, A.K. Yadav, Confabulation of different IoT approaches with and without data compression, Comput. Integr. Manuf. Syst. 28 (11) (2022) 963–981.