Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Development of a cloud-assisted classification technique for the preservation of secure data storage in smart cities

Ankit Kumar[1], Surbhi Bhatia Khan[2,3,4*], Saroj Kumar Pandey[1], Achyut Shankar[5], Carsten Maple[6], Arwa Mashat[7] and Areej A. Malibari[8]

Abstract

Cloud computing is the most recent smart city advancement, made possible by the increasing volume of heterogeneous data produced by apps. More storage capacity and processing power are required to process this volume of data. Data analytics is used to examine various datasets, both structured and unstructured. Nonetheless, as the complexity of data in the healthcare and biomedical communities grows, obtaining more precise results from analyses of medical datasets presents a number of challenges. In the cloud environment, big data is abundant, necessitating proper classification that can be effectively divided using machine language. Machine learning is used to investigate algorithms for learning and data prediction. The Cleveland database is frequently used by machine learning researchers. Among the performance metrics used to compare the proposed and existing methodologies are execution time, defect detection rate, and accuracy. In this study, two supervised learning-based classifiers, SVM and Novel KNN, were proposed and used to analyses data from a benchmark database obtained from the UCI repository. Initially, intrusions were detected using the SVM classification method. The proposed study demonstrated how the novel KNN used for distance capacity outperformed previous studies. The accuracy of the results of both approaches is evaluated. The results show that the intrusion detection system (IDS) with a 98.98% accuracy rate produces the best results when using the suggested system.

Keywords Cloud, Smart cities, Data protection, Intrusion detection, Machine learning

*Correspondence:
Surbhi Bhatia Khan
surbhibhatia1988@yahoo.com
[1] Department of Computer Engineering & Applications, GLA University, Mathura, India
[2] Department of Electrical and Computer Engineering, Lebanese American University, Byblos, Lebanon
[3] Department of Data Science, School of Science, Engineering and Environment, University of Salford, Manchester, UK
[4] Byblos, Lebanon
[5] WMG, University of Warwick, Coventry CV4 7AL, UK
[6] Secure Cyber Systems Research Group (SCSRG), WMG, University of Warwick, Coventry, UK
[7] Faculty of Computing and Information Science, King Abdulaziz University, Rabigh, 25732, Saudi Arabia
[8] Department of Industrial and Systems Engineering, College of Engineering, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

Introduction

Cloud computing can be used to explore the full potential of smart city services, which are supported by highly inventive and scalable service platforms. Smart cities require a decentralized cloud-based platform and an open-source network to be implemented. Multi-sensor apps can perform complex big data processing using dispersed sensor networks thanks to Internet of Things features included in the cloud platform [1]. The Indian government has different plans for implementing the smart city objective in different cities, depending on the level of development required. India is transforming both rural and urban areas into smart cities in order to improve the quality of life and communication between the government and its citizens. Many factors influence

the growth of a smart city, including the facilitation of multiple land uses, the provision of adequate housing for all, the encouragement of multiple modes of transportation, the creation of citizen-friendly and cost-effective governance, and the provision of a distinct character for the city. A cloud-assisted categorization strategy for secure data storage preservation in smart cities is a highly efficient and secure approach to organizing and preserving data that is collected from various smart city devices such as sensors and cameras. This strategy involves leveraging cloud computing technology to store and process data, along with a classification algorithm to sort the data into specific categories based on certain criteria.

The primary objective of this strategy is to provide a secure and scalable solution for managing the vast amounts of data generated by smart city devices. With the use of cloud computing, data can be centrally stored and processed, making it more straightforward to manage and analyze. Furthermore, the classification algorithm ensures the efficient categorization of the data based on its content, which facilitates easier storage and retrieval.

This technique has the potential to enhance the effectiveness of smart city initiatives by enabling better data management and analysis. The centralized storage and analysis of data can provide insights that can inform decision-making and lead to improved services for residents. Additionally, the use of a classification algorithm streamlines the organization of the data, making it easier to access and retrieve specific information when needed.

Cloud computing [2] provides a large platform for smart cities by providing domain-specific applications with the services they require, driving the design of all system components, and determining the majority of technical choices for everything from intelligent devices and sensors to middleware and computing infrastructure.

Figure 1 depicts the entire datamining process, from data storage to data analytics. However, when the amount of data stored becomes extremely large, handling and managing it becomes extremely difficult. Structured databases and database management systems are thus created to address these issues. Efficient database management systems are required for retrieving specific information from large amounts of aggregate data. Because database management systems are widely used, gathering all types of information is simple [3]. Data warehouses collect and store information from various sources. Data mining is a powerful tool for many businesses because it reduces the amount of information available in data warehouses. To differentiate data mining tools, an automated analysis process is used. As a result, new information can be discovered using historical data. As a result, at a specific time, a large set of data is analyzed using data mining.

This method is used to analyses various fields or variables in small data samples. This approach provides simple and effective solutions for performing relatively simple data analysis. Essential data that is present in an unorganized manner can be discovered by effectively using data mining. Data mining tools are used to discover previously unknown patterns in databases. Fraudulent credit card transactions are detected, and anomalous data identification is performed, resulting in pattern discovery issues. The representation of fundamental data entry errors is done here. For presenting the final results, the network supervisor domain experts are presented in an understandable human form. They extract predictive information from applications using highly efficient data mining tools. Text reports, scientific data, or satellite images can all be valuable sources for extracting information. It is not enough to simply retrieve information in order to make decisions. To improve decision-making, new methods for dealing

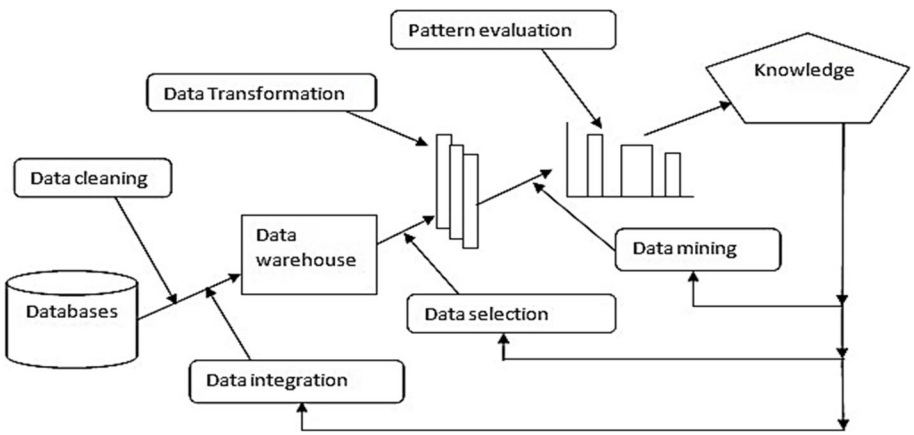

Fig. 1 Data mining process

with primarily collected data are developed. This technology can extract the essence of stored information, discover patterns in raw data, and perform automatic data summarization [4].

Motivation

In order to effectively manage network traffic, it is necessary to categorize input data requirements into defined classes through classification methods. The behavior of data flow on the network must be analyzed, and traffic must be classified into attacker and non-attacker categories. To achieve this, a wire shark dataset is utilized, which goes through three critical steps. The first step is data preprocessing, which eliminates redundancy in the data [5]. The next step is clustering, which groups data into clusters based on their similarity and dissimilarity. The center point of each cluster is determined using the k-means clustering approach. The Euclidean distance is then computed to describe the distance between each data point and the center point. Finally, a classifier is used to categorize the input data based on polarity, resulting in accurate classification while reducing execution time. Future work could focus on enhancing the classification accuracy by incorporating a hybrid classifier or by exploring different classification algorithms [6, 7]. Additionally, the study acknowledges the limitations of only considering technological means of addressing network security, and suggests that legal and institutional frameworks should also be taken into account.

Major contributions

This research work presents several significant contributions to the field of intrusion detection in cloud storage:

i) A novel intrusion detection method using the K-Nearest Neighbor (KNN) classification algorithm has been introduced. This method analyzes the unusual patterns of activity in the network and identifies and isolates abnormal nodes.
ii) A framework model has been proposed that utilizes a modified version of the KNN algorithm for classifying network traffic. The model applies a predicting algorithm to improve the accuracy of the classification process.
iii) Machine learning techniques have been employed to evaluate the effectiveness of the proposed algorithm and achieve the desired results.
iv) The outcomes of the proposed approach have been analyzed and compared with those of existing methods in terms of various parameters such as accuracy, execution time, and information retrieval metrics.

Organization of paper

The subsequent sections of this paper are structured as follows: in section 2, presented the Literature review, and section 3 describes the Proposed work along with the methodology. Moving on, Section 4 examines the analysis of the results. Finally, in Section 5, the conclusions of the paper and future work are presented.

Literature review

In this section, we have reviewed the existing work & methods completed by the different researchers.

The fundamental research issue surrounding the cloud is ensuring the order of clients' data in the cloud. Customers' various data is stored by big data storage providers; this should be confirmed. Distributed computing has steadily advanced in information technology and will continue to shape I.T. organizations in the coming years.

Cloud is also facing significant difficulties. Ensure that appropriate physical, canny, and staff security controls are in place, especially when collecting cloud data [8]. Furthermore, when moving such massive amounts of data, the data organization may not be reliable. This territory depicts the investigation work related to the issue space of ensuring data security in cloud storage. A brief report summarizes the research conducted by several scientists in the field of sickness prognosis. According to [9], the primary goal of the data security model is to detect attacks in the rail transportation sector. An expert attack detection system known as the BAS was developed to detect assaults and reduce their impact on the subway's environment control subsystem. Expert systems enable the detection of unauthorized operations and attacks, as well as the inference engine and knowledge base. There are blacklist and allow list regulations included, which can be used to prevent unauthorized attacks. The regulations provided extensive protection for the subway system's environment control system's data security. This method protects the data of several subway subsystems. This technology is currently being tested due to a number of limitations. However, IDSs can be deployed in urban areas using big data principles.

The authors of [10] discussed internal IDs and IDS models. Real-time forensic algorithms and data mining techniques are used in these models. Data mining techniques were demonstrated to aid in cyber investigation and attack detection. Several analyses from different researchers were used to provide a variety of methods for detecting assaults, which were reported in this paper. The evaluation of this work was beneficial in reaching a satisfactory conclusion. The proposed method improved the precision and increased the number of new discoveries by up to 95%. Existing methods, by contrast, have a 90%

accuracy and discovery rate. Based on these findings, it was obvious that the proposed method outperformed previous algorithms in terms of precision and intrusion detection.

The study conducted by the author of this paper [11] aimed to investigate an intrusion detection algorithm capable of classifying a large percentage of potential attacks as true or false without the need for operator input [12]. The proposed algorithm was developed using immunology stimulation rules and a Negative Selection algorithm. To achieve this, the co-stimulation system and a two-tier negative selection technique were employed. The primary objective of the system was to minimize detection errors while reducing the need for human intervention. Through the proposed MNSA algorithm, the study was able to detect around 34% of all attacks without the need for non-self-information. Moreover, the algorithm confirmed over 90% of the recognitions that did not require additional data or an operator unit. This implies that the proposed algorithm has the potential to significantly reduce the workload of network administrators and enhance the efficiency of network security.

However, there are still some limitations to the proposed algorithm that need to be addressed. For instance, the algorithm's accuracy and performance might be affected by the variation in the types of network traffic. Therefore, further research is required to validate the effectiveness of the algorithm in different network environments. Additionally, the algorithm's ability to detect unknown or zero-day attacks needs to be evaluated to determine its overall reliability in real-world scenarios. To prevent lung cancer, the author [13] proposed a brand-new clustering technique called Foggy K-means algorithm. To prevent lung cancer, a suggested strategy and powerful analytical ability were offered. This study compared the proposed method with the traditional K means algorithm. By comparing results, the cluster authenticity criteria were shown to better suit the proposed method [14]. Field experts could use these findings to create more robust clusters for prediction. The harmful effects of smoking, tuberculosis, radiation produced by various industries, and radioactive materials may all be linked to various illnesses. The results of the proposed clustering technique could be used to category lung cancer patients in future research. This method will identify the factors that have a significant impact on lung cancer.

According to the paper [15], various prediction instruments were used for clustering. This study proposed a novel modified approach for climate prediction called K-mean clustering generic methodology. The goal of this project was to measure the level of pollution in the air. A dataset from the state of West Bengal was used for this purpose. Using the peak mean values of the clusters, a climate group catalogue was created. The K-Means clustering algorithm was used in the air pollution data suite. Climate groups were described using various clusters. The term "modified K" denotes that the algorithm validated the new data and classified it into accessible clusters. The proposed method predicts information on upcoming climate conditions. West Bengal state weather forecasting data was included in the data set. The effects of air pollution could be mitigated with the help of this data set. The modelled estimates accurately predicted climate conditions. Finally, the authors conducted various tests to validate the proposed algorithm's accuracy.

The article's author [16] describes the Student Achievement Analysis System (SPAS), which tracks students' academic performance at a specific institution. This work's proposed approach included a forecasting model. This forecasting model could predict the performance of students in a specific course. This course sequentially assisted professors in recognizing poor student performance. These students were predicted using the proposed method. Some data mining rules were used to forecast student performance. A data mining technique known as classification was used in this work. This technique classified students based on the grades they received.

According to the authors of the paper [17], predicting share profit is an important topic in data analysis and prediction. It was assumed that the historical primary data had some analytical relationship with future share profits. The information retrieved from the past worth of these shares was used to decide the selling and purchasing of shares in this work. As a result, those who invested in the stock market benefited from this strategy. A classification model known as a decision tree was used in this work.

To predict the analysis problems, the researcher used the k-means algorithm to present the results based on accuracy [18]. For this purpose, both natural and synthetic datasets were used. K-Means was a clustering method. The primary goal of this algorithm was to divide n patterns into k clusters. Every pattern was linked to the cluster with the lowest mean. Each cluster was assigned a random number of clusters, k. Every integer was given a random start value. The proposed technique was used to category the collection of items based on their characteristics. These objects were divided into K groups. To group objects, the sum of the squares of distances between them was minimized. For this, the Euclidean distance formula and the corresponding cluster centroid were used. Clustering produced effective results with the highest accuracy and robustness, according to the tests.

The author briefly explains the concept of clustering in this work. Clustering divided the data into clusters of

similar entities [19]. Objects in each cluster were similar. These objects, however, were distinct from those in other clusters. K-means is a well-known clustering algorithm. This algorithm was widely used in data clustering. However, this algorithm is computationally expensive. The choice of initial centroids had a significant impact on the quality of the final results. This paper proposed a novel approach for improving the algorithm's competence and productivity. The technique presented here reduces the difficulty and time required for mathematical computation. Furthermore, the proposed technique preserved the ease of use of the k-means algorithm. The proposed solution also addresses the issue of the dead unit.

The article [20] discusses research on methods for classifying and predicting non-linear datasets. And it has been stated that, when compared to other approaches used for prediction and classification, the neural network approach is generally regarded as the best classification method. The B.P. algorithm is the most effective classifier of an artificial neural network because it uses the updating approach of weights. Faults are also propagated backward using this method. This method is constrained by local minima solutions. This study solves the problem by employing an effective modified technique that improves accuracy and is used in a variety of future prediction applications.

The study's authors proposed classification methods for risk prediction, pattern recognition, and data mining in clinical cardiovascular medicine [21]. The data has been modelled and classified using a data mining technique known as categorization. Unfortunately, conventional medical scoring methods can only be used up to a point due to the linear combination of elements in the input set. As a result, non-linear complex interaction modelling is not used in medicine. Classification methods are used to overcome this limitation because complex non-linear correlations between dependent and independent variables can be discovered. Furthermore, it can identify any and all possible links between various prognostic indicators.

The study's author [22] proposes two methods for selecting features from the dataset: SVM-RFE and gain ratio. Depending on the circumstances, the healthcare industry has a wealth of data that must be mined for hidden patterns. Data mining techniques in this field are required for optimal judgement. The features saved in the proposed method can be used with the Random Forest and Nave Bayes algorithms. The obtained results can be used to improve the procedure's performance level. Each factor is assigned a specific importance rating using this method. Experiment results confirmed that the proposed method achieves the highest precision with the least amount of computing effort.

The Author [23] discussed the dual issues of privacy and security in a big data-enabled cloud environment in this work. The three methods of big data management discussed in this study are outsourcing from data owners, sharing with data consumers, and cloud-based management. We advocated for the implementation of the SHA3 hashing technology, which generates a hash of user information and stores it in the Trust Center, as a means of providing secure user authentication of Data Owners and Data Users. The data's owners securely transmit it to the cloud server. When data is compressed using the LZMA method, big data-enabled cloud storage becomes more efficient. Finally, we used SALSA20 Encryption Map Reduce to accelerate the encryption and decryption processes. After encryption, the data is uploaded to a remote server.

While cloud computing is relatively [24] mature and its potential benefits well understood by individual, industry and government consumers, a number of security and privacy concerns remain. Unsurprisingly, designing cryptographic solutions to ensure the security of cloud services and the privacy of data outsourced to the cloud remains an ongoing research area. This paper provides a critique of the wide range of cryptographic schemes designed for securing sensitive data in the cloud computing environment, as well as outlining the research opportunities in the use of cryptographic techniques in cloud computing.

Cloud storage systems are increasingly turning to NoSQL [24] database management systems (DBMS) due to their superior availability and performance compared to traditional DBMSs. However, some NoSQL DBMSs sacrifice consistency guarantees for performance gains by using eventual consistency, where an operation is confirmed without checking all nodes. Different consistency levels can be adopted, affecting system behavior. Therefore, it's crucial to assess system design considering distinct consistency levels to develop cloud storage systems. This study proposes an approach using reliability block diagrams and generalized stochastic Petri nets to evaluate availability and performance of cloud storage systems with redundant nodes and eventual consistency based on NoSQL DBMS. The experiment shows that system configuration can cause unavailability from 1 s to 21 h in a year, and performance can decrease by up to 17.9%.

This paper [25] investigates the problem of efficient data integrity auditing supporting provable data update in cloud computing environment. It introduces an efficient outsourced data integrity auditing scheme based on the Merkel sum hash tree (MSHT). The scheme could meet the requirements of provable data update and data confidentiality without dependency on a third authority.

This paper [26] introduces a threefold methodology to improve the trade-off between I/O performance and capacity utilization of cloud storage for CDS services. This methodology includes:

i) Definition of a classification model for identifying types of users and contents by analyzing their consumption/ demand and sharing patterns,
ii) Usage of the classification model for defining content availability and load balancing schemes, and
iii) Integration of a dynamic availability scheme into a cloud-based CDS system.

This paper [27] presents a comparative and systematic study of leading techniques for secure sharing and protecting the data in the cloud environment. It discusses the functioning, potential, and achievements of each solution and provides a comparative analysis. The applicability of the techniques is discussed as per the requirements and the research gaps along with future directions are reported in the field.

This paper [28] discusses a new generation cloud storage system that integrates distributed storage technology. It is designed to support all kinds of OLTP or OLAP business applications and to solve the problems of data security and smooth storage expansion.

The identity of the Data [29] User making a request for data must be confirmed by the Trust Center before the request can be fulfilled. To read the specified data file, the secret keystream is applied. We looked at two methods, clustering with DBSCAN and indexing with Fractal Index Tree, for big data management in the cloud. The proposed SADS-Cloud technique was developed for the E-healthcare application, evaluated, and compared to other approaches based on a number of parameters, including information loss, compression ratio, throughput, encryption time, decryption time, and efficiency.

The lack of consideration for the influence of the suggested approach on energy consumption and environmental sustainability is a limitation of the literature review. While using cloud computing for data storage and processing has advantages such as scalability and simplicity of management, it also consumes a lot of energy and has a detrimental influence on the environment. As a result, future research might concentrate on creating and assessing strategies that combine the advantages of cloud computing with energy efficiency and ecological concerns. Another possible research gap is the requirement for a more thorough evaluation and testing of the suggested approach on real-world smart city datasets. While the research exhibits promising findings on a simulated dataset, the suggested technique's performance may vary in different smart city scenarios with variable data qualities and volume.

As a result, future research may include testing and evaluating the suggested approach on a variety of real-world smart city datasets to assess its efficacy and applicability in various scenarios. The study focuses mostly on the technological components of the suggested method, with less emphasis placed on the social and ethical consequences of using cloud computing and data categorization in smart cities. Future study might look at the social and ethical implications of using such approaches in smart city settings, such as privacy, data ownership, and responsibility. Comparative study of exiting work shown in Table 1.

Proposed work

Prediction analysis is the process used to forecast potential future outcomes based on present data. Prediction analysis's foundation is clustering and classification. Clustering and classification are the two parts of the prediction analysis process. The cluster head in this research is constructed using the k-mean clustering technique. The output is used as a classification input by the SVM classifier.

The intrusion detection system in this study makes use of a KNN and an SVM model to carry out its operations. There are three benefits to the system:

Table 1 Comparative analysis of exiting work

Attack model	[8]	[13]	[15]	[20]	[23]	[24]	[30]
Insider attack	Yes	No	Yes	Yes	Yes	Yes	Yes
Mutual authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Offline password guessing attack	Yes	Yes	Yes	Yes	No	No	Yes
Perfect forward secrecy	Yes	No	Yes	Yes	No	Yes	Yes
Replay attack	Yes	Yes	No	Yes	No	Yes	Yes
User impersonation attack	No	No	No	No	No	No	Yes
Known-key attack	No	No	No	No	No	Yes	Yes

1. First, the value of has little impact on the final findings; second, the cutoff value used to designate the anomalous node is easy to estimate; and third, the process moves quickly and produces accurate results.
2. There are two inputs needed for the KNN classifier: value and the cutoff value. This represents the total number of nodes that are quite close together. The cutoff value is the criterion used to rank the outliers among the nodes. The following terms are defined to help clarify this method's procedure:
3. The node's feature vector is composed with Network, S is the collection of all nodes in the network, whether pathological and normal.
4. The distance between two separate nodes and is their Euclidean distance, denoted by eudis.
5. The distance function of a node is the value obtained by adding the Euclidean distances of all of its neighboring nodes.

Research methodology

The prediction analysis is carried out in this study. Based on the existing dataset, the prediction analysis can forecast future opportunities.

The first section of this research looks at how the KNN classification approach can be used to solve the problem of intrusion detection in wireless sensor networks. An intrusion detection system based on the KNN algorithm is evaluated for parameter selection and error rate in order to distinguish abnormal nodes from normal ones. We decided to put the intrusion detection system through its paces to see how effective it was. The terminal device's physical foundation is made up of both wireless sensor nodes and a wired network card. The wireless sensor nodes used to monitor network activity and propagate blacklists are manufactured by Ningbo Zhongke Integrated Circuit Co., Ltd. Terminal hardware allows for the detection of anomalies in control systems, network traffic, node anomaly evaluation, and attack resolution. The software stack includes TinyOS, an embedded operating system, and the AVRStudio IDE. A serial communication aid is used to exchange control data messages.

The intrusion detection system is a complex system that consists of various components, including a wireless network interface module (WAN IM), a data storage module, an analysis and judgment module, and an intrusion reaction module. The WAN IM is installed on the wireless sensor nodes to collect raw data, which is then stored in the data domain by the data storage module. The data is then used in the evaluation and analysis phase. The analysis and judgment module reads the test settings and data from the data storage module to analyze and draw conclusions based on the data. This module also updates the intrusion response module. The intrusion response module plays a critical role in notifying the wireless network interface component of the malicious nodes that need to be blocked. Once a blacklist containing the abnormal nodes has been broadcast throughout the network, normal nodes will stop receiving and relaying RREQ signals from the abnormal nodes. This is because any unusual node will be prohibited from further communication. At the same time, the blacklist will be distributed to other nodes to assist in responding to a flooding attack. To improve the accuracy of the system, we trained the model for a total of one thousand cycles. The training process allows the system to learn from the data and improve its performance over time. This intrusion detection system is a crucial tool for maintaining the security of wireless sensor networks and ensuring the safe and efficient operation of smart city applications.

k-mean is a clustering algorithm. Similar and dissimilar data are grouped using this method based on their similarities. In the k-mean clustering, the dataset is considered by the k-mean method. The arithmetic mean is computed using this dataset. The arithmetic mean represents the dataset's focal point. Starting from the center, the Euclidian distance is calculated [31]. The points that are comparable and different are also divided into distinct groups. In this study, the Euclidian distance is measured dynamically. This effect improves the clustering accuracy. To measure Euclidian distance dynamically, this study employs a technique known as backpropagation. This method clusters uncluttered points and improves the clustering accuracy.

Pre-processing

In this step, the data is provided as input. Missing values are depicted in the cleaned data. In this step, the redundant values are removed. In this step, the standard deviation, mean values, and so on are calculated.

Phase of prediction

In this step, the division of the input dataset results in the generation of training and testing sets. As shown in Fig. 2, we divided the dataset in the tanning set data into two parts: the first portion (70%) is used as a tanning set, and the remaining 30% is used as a testing test.

The prediction analysis is performed using the KNN classification model. This classifier accepts training and testing data as input. Predicted data is the output that is obtained. The K-Nearest Neighbor (KNN) algorithm is

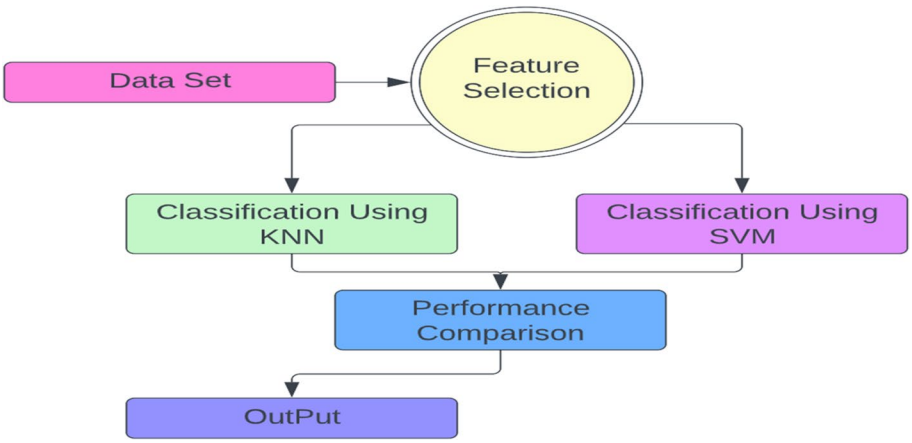

Fig. 2 Work flow diagram

Table 2 List of mathematical notations used

$f_{en}(t)$	Objective function representing Residual energy
$f_{PDR}(t)$	Objective function representing PDR
N^{i}	Ith Node
$\rho_{i,j}$	Pheromone information for the link (i,j)
Nei_i	Neighbor set for ith Node
q	Number of elements in the cluster produced from k-means algorithm
$E(t)$	Pheromone evaporation coefficient
p	Number of feasible paths from source to destination node
Seq_s	Set of sequence numbers received at source node
ct	Centroid of the cluster formed using k-means algorithm
k	Number of neighbors
P^{r}	Number of packets received
$\tau_{i,j}$	Heuristic information for the link (i,j)
T_p	Predicted time period for sequence numbers
s	Slope of Line of Best Fit
$C.E$	Energy consumed by the node
$\rho_{i,j_k}(t)$	Pheromone level
$R.E$	Residual/Remaining
c	Number of partitions required from k-means clustering
P^{d}	Number of packets dropped

a simple method. KNN is a non-parametric supervised learning approach since it doesn't make any assumptions about the underlying data distribution. This technique categorizes the patterns based on neighboring training patterns in the feature space. The labels of the training pictures are used to store the feature vectors throughout the training process. The unlabeled question point in the categorization is distributed in the direction of its k-nearest neighbors' labels. The item is chosen via majority vote sharing based on the labels of its k closest neighbors. The classification of the object is done successfully in this algorithm. The nearby object class in the scenario when k = 1. When there are only two classes, k is an odd integer. When multiclass categorization is used, there can be a tie if k is an odd whole number [32]. Table 2 contain all the mathematical notation which is used in this paper.

This classifier's primary goal is to categorize patterns according to the majority class of their closest neighbors.

$$Class = argvmax \sum (Xi, yi) \in DzI(v = yi) \quad (1)$$

Variable v in the equation above represents the class label [33]. The class label for its closest neighbors in this equation is y_i. The variable I represents the indicator function. In this function, the value "1" is returned in case of an actual argument. If the opposite is true, "0" is returned. As a result, the patterns are allocated to its K closest neighbors' class. A collection of labeled objects, a distance or similarity measure, and other key elements of this approach are identified. These components calculate the separation between objects and their closest neighbors. The value of k serves as a proxy for the distance. The identification task can be successful by choosing a suitable similarity function and values for parameter k.

Previous algorithm

As observed in the IDS packet transformation, the categorization difficulties are solved using supervised machine learning algorithms. Your data is transformed using a method known as the kernel trick, and based on these alterations, it determines the best cutoff between the possibilities.

Algorithm 1. IDS packet transformation

```
Input: Route packet for node 1 to n
Outputs. IDS Packet Classification
Initialize dataset
        DS = {Compute the closest classes}
while there are violating points then do
        Find a violator
        DS= DS    violator then compute if any αp < 0   then   add c+ S then
            DS= DS / p
            repeat   all such points are pruned
        end if
end while
```

Proposed algorithm

The k-mean clustering process results in some points being left unflustered, which reduces accuracy. When using k-mean clustering on the dataset, the whole dataset, including all instances, is utilized as input. The whole dataset was divided into groups of similar kind using K-mean clustering. The results of the k-mean clustering technique will be used as input for the SVM classifier, which may categorize data based on hyperplanes [34]. The k-mean technique will be enhanced for clustering in this research. The classification process will use the clustering result as input, which improves the prediction analysis accuracy.

As observed in the IDS packet transformation, the categorization difficulties are solved using supervised machine learning algorithms. Your data is transformed using a method known as the kernel trick, and based on these alterations, it determines the best cutoff between the possibilities.

$$\overline{v}_i = \frac{\delta x}{\delta t}\left(\frac{n!}{r!(n-r)!}x^{\gamma} + \mu(x)\right) \tag{2}$$

Determine the distance in Euclidean space between the remaining data items and the first grouping centers U_i:

$$RSU_i = R \cdot M^N \cdot \sum_{i=1}^{n} \left(V_i - \overline{V}\right)^2 \tag{3}$$

$L(\beta)$ is the Levy distribution function's probability density function?

$$L(\beta) = \prod \beta(V_i + \theta) + \eta \tag{4}$$

Determine the judgment value for each location's odor concentration:

$$\text{smelli} = \sum_{m\neq a, n\neq b}^{i} \left[\alpha_m^2(t)\right]^i \left[\beta_n^2(t)\right]^j \tag{5}$$

The text input is altered by the language model and converted to a vector, where cosine similarity G is a popular metric of similarity (J):

$$G(J) = j\frac{\partial \gamma}{\partial j} + \frac{1}{n}\sum_{i=1}^{n} X_i Y_i \tag{6}$$

The most extensively researched and used technique for unsupervised learning problems is cluster analysis [35]. The class approach separates a data set into multiple different subsets called "class clusters," each with its own clustering center. It is determined by how similar each sample is to every other sample in a data set. Throughout the clustering method, only the cluster structure is formed automatically. Each node in a network creates its own cluster, with one node serving as the cluster's leader. Data from the cluster's other nodes is delivered to the cluster head, who aggregates the information, adds signal processing to it, and delivers it to the distant base station. As a result, operating as a cluster head node requires much more resources than

serving in another role. As a result, if the node functioning as the cluster's head dies, all nodes in the cluster lose their ability to communicate with one another. in this paper is that the authors have chosen two types of malware that make significant modifications to the guest operating system, which makes them relatively easy to detect. The author believes that these types of malware would also be easily detected by signatures.

However, the reviewer suggests that in order to prove the effectiveness of the approach proposed by the author, they should have chosen more covert malware, such as rootkit kernels, which are known for being particularly difficult to detect. By choosing such malware, the author could demonstrate whether their detector is able to detect subtle deviations and thus be more effective in detecting sophisticated attacks.

Algorithm 2. Cluster the Node

Input: Block Data as $M = (m_1, m_2, \ldots, m_n)$

Output: K cluster of IDS

```
Assigns the number of nodes in the input and output layer
Assign and initialize the value of node weight between -1 and 1 repeat
    For value in tanning set
        Compute the cluster in the network
            for every layer in the network & node
                Compute the weighted sum for every node
                    Compute the threshold and add it to the sum
                        Compute the Activation function for every node
            end
    end
    for every node in the output layer
        Compute Signal Error Value
    end
    Repeat for all hidden layers in the network
        Compute Signal Error Value
            Update the weighted sum for every node
                end
            end
        end
        Error Function is call to compute their value
    while ((max iterations < specified value) AND
(Error Function > than specified value))
```

The arithmetic mean of the complete data set is taken to measure the center points in this approach. The points with similar values are grouped in an individual cluster, while others are grouped in a different cluster. Consider the problem of clustering a set of n objects $I = \{1 \ldots, n\}$ into K clusters. For each object $i \in I$, we have a set of m features $\{x_{ij} : j \in J\}$, where x_{ij} describes the j the features of object i quantitatively. Let $x_i = (x_{i1}, x_{im})^T$ be the feature vector of the object i and $X = (x_1, \ldots, x_n)$ be the feature matrix or data set.

As an optimization problem that minimizes the following clustering objective function, the clustering job may be restated:

$$mJ(U, V) = \sum_{k=1}^{K} m* \sum_{i \in I} m*u_{ik} \parallel x_i - v_k \parallel_p^p \tag{7}$$

under the following constraints:

$$\sum (k = 1)^K u_{ik} = 1, u\_ik \in \{0, 1\}, \forall i \in I, k = 1, \ldots, K, \tag{8}$$

where p=1,2. For $k = 1, \ldots, K, v\_k \in R^{mis}$, the kth cluster prototypes, and for every $i \in I, u\_ik$ identifies whether the item I is a member of the kth cluster. For p=1 and p=2, the clustering issue may be solved effectively using the K-median and K means methods. Let the cluster prototype matrix be in the following $V = [v_1, \ldots, v_K] \in R^{m \times K}$, and the membership matrix $U = [u_1, \ldots, u_n] \in R^{K \times n}$, where $v_i = (v_{i1}, \ldots, v_{im})^T and\ u\_i = (u_{i1}, \ldots, u_{iK})^T$

Both algorithms solve the clustering problem in iterative ways as follows:

cluster prototypes $\{v_k^t : k = 1, \ldots, K\}$.

Step 2. Let $t = t + 1$, and update the membership matrix U^t by fixing the cluster prototype matrix $V^{\wedge}(t - 1)$. For any $i \in I$, randomly select $k^t* \in argmin\{\parallel x_i - v_k^{t-1} \parallel_p : k = 1, \ldots, K\}$, and set $u_{ik^*}^t = 1$ and, for any $k \neq k^t*, set\ u_{ik}^t = 0$.

Step 3. Update the cluster prototype matrix V^t by fixing the membership matrix U^t. When p=1, for any $k = 1 \ldots, K$ and $j \in J, set\ v_{kj}^t$ as the median of the jth feature values of these objects in cluster k. When p=2, for any $k = 1, \ldots, K$, set v_k^t as the centroid of these objects in cluster k; that is, $v_k^t = \left(\frac{1}{\sum_{i \in I} i} u_{ik}\right) \sum u(i \in I) u_{ik} x_i$.

Step 4. If, for any $i \in I$ and $k = 1, \ldots, K$, we have $u_{ik}^t = u_{ik}^{t-1}$, stop and return to U and V; otherwise, go to Step 2.

Setup the proposed protocol

Each participant, whether CSPs, clients, or auditors $P \in \{CSP$, client, auditor $\}$ carries out Key Gen to acquire skP and vkP. e client takes $s + 1$ samples from the random elements $a_1, 2, ..., s, x, Z, q$ and computes random elements $\alpha_1, \alpha_2, \ldots, \alpha_s, x \in Z_q$ and computes the value of $g_1 = g^{\alpha_1}, g_2 = g^{\alpha_2}, \ldots, g_s = g^{\alpha_s}, y^{'} = g^x \in G$. Now a random element $\lambda \in G$, and the secret key and a public key, which are donated with $sk = (\alpha_1, \alpha_2, \ldots, \alpha_s, x)$ and $pk = (g, \lambda, g_1, g_2, \ldots, g_s, y)$.

Store protocol

Data File in the different blocks as $M = (m_1, m_2, \ldots, m_n)$ And every block contains different s sectors in the form of $m_i = m_{i1} m_{i2} \cdots \parallel m_{is} (1 \leq i \leq n)$ where sector $m_{iz} \in Z_q (1 \leq z \leq s)$, denotes concatenation. Client first computes $h_i = H_2(m_i) (1 \leq i \leq$ n) from the data block on top of the ordered hash values of node w_i stores the corresponding hash value h_i Based on g, λ, and secret key sk, the client computes the value.

M and T are then deleted from the local storage of the client's computer. Only metadata is maintained. Time-dependent pseudo-randomness generated by the Bitcoin blockchain is used to produce periodic challenges. A hash value hash of the latest block that has arrived since time t in the Bitcoin block chain is obtained by entering the time t. A pseudo-random-bit generator C = {**B.I.**, F, l}, where C denotes the auditor's checking policy. It invoked on the input $h^{(b)(b)}$ to receive a random bit by selecting a keys pair $k_\pi^{(b)}, k_f^{(b)}$. Then auditor generates a challenge $Q^{(b)} = \left\{b, k_\pi^{(b)}, k_f^{(b)}\right\}$ And sends it to CSP [30].

The challenge $Q^{(b)}$ CSP Computed the indices and coefficients by using the equations:

$$\begin{aligned} i_\eta &= \pi_{k_\pi^{(b)}}(\eta), \\ a_\eta &= f_{k_f^{(b)}}(\eta) (1 \leq \eta \leq l) \end{aligned} \tag{9}$$

Then, CSP validates the proof of data to check the integrity of the challenged blocks by the following equations:

$$\begin{aligned} (b)_z^{(b)} &= \textstyle\sum_{\eta=1}^{l} a_\eta m_{i_\eta z} \in \mathbb{Z}_q,\ 1 \leq z \leq s, \\ \sigma^{(b)} &= \textstyle\prod_{\eta=1}^{l} \sigma_{i\eta}^{a_\eta} \in G. \end{aligned} \tag{10}$$

The proof $\rho^{(b)} = \left\{\mu_1^{(b)}, \mu_2^{(b)}, \ldots, \mu_s^{(b)}, \sigma^{(b)}\right\}$ the auditor verifies the correctness of $\rho^{(b)}$. It verified the indices and coefficients by the value with T as using the equation:

$$h^{(b)} = \lambda^{\sum_{\eta=1}^{l} a_\eta h_{i\eta}} \in G \tag{11}$$

Third, the auditor verifies the proof $\rho^{(b)}$ by checking the following equation:

$$e\left(\sigma^{(b)}, g\right) \stackrel{t}{=} e\left(h^{(b)} \cdot \prod_{z=1}^{s} g_z^{\mu_z^{(b)}}, y\right) \tag{12}$$

The auditor verifies that the challenged data blocks are intact if the equation holds. Auditor saves a log entry to document their auditing of the data:

$$\boldsymbol{L}^{(\boldsymbol{b})} = \left\{ \boldsymbol{t}, \boldsymbol{Q}^{(\boldsymbol{b})}, \boldsymbol{h}^{(\boldsymbol{b})}, \boldsymbol{\rho}^{(\boldsymbol{b})}, \mathbf{Sig}_{sk_{\mathrm{CSP}}}\left(\boldsymbol{\rho}^{(\boldsymbol{b})}\right) \right\} \tag{13}$$

A random is chosen by the client from the subset $\boldsymbol{B}$ of indices of Bitcoin blocks and transmitted to the auditor. Then auditor receives the value of $\boldsymbol{Q}^{(\boldsymbol{b})}, \boldsymbol{h}^{(\boldsymbol{b})}$, and $\boldsymbol{\rho}^{(\boldsymbol{b})}$ From **log** file $\boldsymbol{\Lambda}$.

$$\begin{aligned} \boldsymbol{h}^{(\boldsymbol{B})} &= \prod_{\boldsymbol{b}\in\boldsymbol{B}} \boldsymbol{h}^{(\boldsymbol{b})} \in \boldsymbol{G}, \\ \boldsymbol{\sigma}^{(\boldsymbol{B})} &= \prod_{\boldsymbol{b}\in\boldsymbol{B}} \boldsymbol{\sigma}^{(\boldsymbol{b})} \in \boldsymbol{G}, \\ (\boldsymbol{B})_{\boldsymbol{z}}^{(\boldsymbol{B})} &= \sum_{\boldsymbol{b}\in\boldsymbol{B}} \boldsymbol{\mu}_{\boldsymbol{z}}^{(\boldsymbol{b})} \in \mathbb{Z}_{\boldsymbol{q}}, 1 \le \boldsymbol{z} \le \boldsymbol{s}. \end{aligned} \tag{14}$$

Challenge index vector is denoted by $\boldsymbol{C} = (\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_{\boldsymbol{c}})$. Now it obtains the corresponding multi-proof $\boldsymbol{\Delta}_{\boldsymbol{p}}$. Then auditor generates the proof of the appointed logs as follows:

$$\boldsymbol{\rho}^{(\boldsymbol{B})} = \left\{ \mathbf{U}_{\boldsymbol{p}}, \boldsymbol{h}^{(\boldsymbol{B})}, (\boldsymbol{B})_{\boldsymbol{1}}^{(\boldsymbol{B})}, (\boldsymbol{B})_{\boldsymbol{2}}^{(\boldsymbol{B})} \ldots, (\boldsymbol{B})_{\boldsymbol{s}}^{(\boldsymbol{B})}, \boldsymbol{\sigma}^{(\boldsymbol{B})} \right\} \tag{15}$$

and sends it to the client with $\mathbf{Sig}_{sk_{\boldsymbol{a}}}\left(\boldsymbol{\rho}^{(\boldsymbol{B})}\right)$.

It was verifying $\boldsymbol{the}\mathbf{Sig}_{sk_{\boldsymbol{a}}}\left(\boldsymbol{\rho}^{(\boldsymbol{B})}\right)$ And invoke the $\left(\boldsymbol{\rho}^{(\boldsymbol{B}}\right.$ hash $\left.^{(\boldsymbol{b})}\right)$ to receive $\boldsymbol{Q}^{(\boldsymbol{b})}$ and indices and coefficients are verified $\boldsymbol{i}_{\boldsymbol{\eta}}, \boldsymbol{a}_{\boldsymbol{\eta}} (\boldsymbol{1} \le \boldsymbol{\eta} \le \boldsymbol{l})$. The client verifies $\boldsymbol{h}(\boldsymbol{B})$.it by using the Eq. 16.

$$\boldsymbol{h}^{(\boldsymbol{B})} \stackrel{\mathbf{n}}{=} \boldsymbol{\lambda}^{\sum_{b\in B}\left(\boldsymbol{l}/\sum_{\eta=1} \boldsymbol{a}_{\eta}\boldsymbol{h}_{i\eta}\right)} \tag{16}$$

The client verifies the secret key $\boldsymbol{sk}$, and the verified $\boldsymbol{h}^{(\boldsymbol{B})}$ as follows:

$$\boldsymbol{\sigma}^{(\boldsymbol{B})}\boldsymbol{n}\left(\boldsymbol{h}^{(\boldsymbol{B})} \cdot \boldsymbol{g}^{\sum_{z=1}^{s} \boldsymbol{\alpha}_{z}\boldsymbol{\mu}_{z}^{(\boldsymbol{B})}}\right)^{\boldsymbol{x}} \tag{17}$$

Equation 17 verifies the client data and node secret key by computing the hash value generated by Eq. 18.

Assuming the calculation above is correct, the customer may be particular that the auditor performed an honest audit of CSP for all previously disputed data blocks appointed by B. The equation's accuracy can be explained as follows:

$$\begin{aligned} \boldsymbol{\sigma}^{(\boldsymbol{B})} &= \prod_{\boldsymbol{b}\in\boldsymbol{B}\boldsymbol{\eta}=\boldsymbol{1}}^{\boldsymbol{l}} \prod_{\boldsymbol{i}_{\boldsymbol{\eta}}}^{\boldsymbol{a}_{\boldsymbol{\eta}}} \\ &= \prod_{b\in B\eta=1}^{l} \prod^{l} \left(\boldsymbol{\lambda}^{\boldsymbol{h}_{i\eta}} \cdot \boldsymbol{g}^{\sum_{z=1}^{s} \boldsymbol{\alpha}_{z}\boldsymbol{m}_{i\eta}}\right)^{\boldsymbol{a}_{\eta}\boldsymbol{x}} \\ &= \left(\prod_{b\in B} \boldsymbol{\lambda}^{\sum_{\eta=1}^{l} \boldsymbol{a}_{\eta}\boldsymbol{h}_{i\eta}} \cdot \boldsymbol{g}^{\sum_{z=1}^{s} \boldsymbol{\alpha}_{z}\left(\sum_{\eta=1}^{l} \boldsymbol{a}_{\eta}\boldsymbol{m}_{i\eta^2}\right)}\right)^{\boldsymbol{x}} \\ &= \left(\boldsymbol{\lambda}^{\sum_{b\in B}\left(\sum_{\eta=1}^{l}(\boldsymbol{a}_{\eta}\boldsymbol{h}_{i\eta})\right)} \cdot \boldsymbol{g}^{\sum_{z=1}^{s} \boldsymbol{\alpha}_{z}\left(\sum_{b\in B} \boldsymbol{\mu}^{(\boldsymbol{b})}\right)}\right)^{\boldsymbol{x}} \\ &= \left(\boldsymbol{h}^{(\boldsymbol{B})} \cdot \boldsymbol{g}^{\sum_{z=1}^{s} \boldsymbol{\alpha}_{z}\boldsymbol{\mu}_{z}^{(\boldsymbol{B})}}\right)^{\boldsymbol{x}} \end{aligned} \tag{18}$$

User define parameters optimizations

In the proposed algorithm, we have used a linear method of classification

$$f_{\mathrm{lin}}(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{w} \rangle_2 + b = \sum_{k=1}^{n} w_k x_k + b\left(\boldsymbol{x} \in \mathbb{R}^n\right) \tag{19}$$

it having a $\boldsymbol{a} \in \mathbf{R}^n$ and $b \in \mathbf{R}$. that are both unknown but constant. During the SVM training process, the classification parameters (also known as level 1 parameters) are calculated. After fine-tuning these settings, the hypothesis function permits binary classification for any x in the range $\boldsymbol{x} \in \mathbf{R}^n$.

$$h(\boldsymbol{x}) := \mathrm{sgn}\left(f_{\mathrm{lin}}(\boldsymbol{x})\right) \tag{20}$$

$\mathrm{sgn}(\cdot)$ is defined as

$$\mathrm{sgn}(a) = \begin{cases} 1 & \text{if } a \ge 0 \\ -1 & \text{else} \end{cases} (a \in \mathbb{R}) \tag{21}$$

In the case when the dataset under examination is not linearly separable, a nonlinear function: ϕ: $\mathrm{R}^n \to D$ is used to map the data to a space D, where $d \in \mathbf{N}$ is the number of dimensions of the space D. is is used as

$$f_{\mathrm{nonlin}}(\boldsymbol{x}) = \langle \boldsymbol{\varphi}(\boldsymbol{x}), \boldsymbol{w} \rangle_D + b = \sum_{k=1}^{d} w_k \varphi_k(\boldsymbol{x}) + b\left(\boldsymbol{x} \in \mathbb{R}^n\right) \tag{22}$$

where $\varphi(\boldsymbol{x})$ which used for training errors $\boldsymbol{\xi}$, then improved classification parameters are derived from

$$\left.\begin{array}{ll} \min\limits_{\boldsymbol{w}\in D, b\in\mathbb{R}, \xi\in\mathbb{R}^l} & \frac{1}{2} \parallel \boldsymbol{w} \parallel_D^2 + C \sum\limits_{i=1}^{l} \xi_i^q \\ \text{s.t.} & y_i \cdot f_{\mathrm{nonlin}}\left(\boldsymbol{x}^i\right) \ge 1 - \xi_i, i = 1, \ldots, l, \end{array}\right\} \tag{23}$$

Results of analysis

The job required extensive meticulous testing from a data mining standpoint. It's also vital to consider the planning and preliminary processing that went into experimenting. This chapter outlines all experimental equipment that will be used to demonstrate the results of a tiny categorization of UCI data set using Python code.

UCI dataset

The University of California School of Information and Computer Science has a substantial collection of datasets that may be used in research projects [36]. According to the kind of machine learning problem, the datasets are categorized. Datasets for classification, regression, recommendation systems, and univariate and multivariate

Table 3 Dataset for classification

Id	Bot name	Packet Send	Sender IP Address	Port	Reciver IP Address	Port
101	spambot malicious download	45912	68.91.226.37	80	172.28.194.173	59594
102	spam bot	45917	172.28.194.173	0	77.91.104.22	80
103	spam bot	45918	172.28.194.173	0	66.200.1.1	25
104	spam bot	45919	172.28.194.173	0	64.180.1.1	25
105	spa m bot	45920	172.28.194.173	0	24.145.1.1	25
106	spam bot	45921	172.28.194.173	0	70.98.1.1	25
107	out2in dns	45922	255.255.255.255	0	255.255.255.255	53
108	compromised_server	45923	172.28.108.88	0	255.255.255.255	53
109	compromised_server	45924	172.28.188.34	0	255.255.255.255	53
110	compromised_server	45925	172.28.18.49	0	255.255.255.255	53
111	compromised_server	45926	172.28.193.152	0	255.255.255.255	53
112	compromised_server	45927	172.28.154.191	0	255.255.255.255	53
113	compromised_server	45928	172.28.140.128	0	255.255.255.255	53
114	compromised_server	45929	172.28.105.141	0	255.255.255.255	53
11–5	compromised_server	45930	172.28.117.175	0	255.255.255.255	53
116	compromised_server	45931	172.28.151.92	0	255.255.255.255	53
117	compromised_server	45932	172.28.186.148	0	255.255.255.255	53
118	failed attack exploit/Hs-asp-overflow	45933	25.178.184.105	0	172.28.20.14	80
119	failed attack exploit/ifs-asp-overflow	45934	25.178.184.105	0	172.28.128.124	80
120	ddos	45937	19.202.221.71	0	172.28.4.7	80
1211	spambot client compromise	45938	201.89.32.16	80	172.28.11.150	57885
122	spambot malicious download	45939	68.91.226.37	80	172.28.11.150	60,695
123	spambot client compromise	45940	201.89.32.16	80	172.28.131.186	49207
124	spam bot	45941	172.28.11.150	0	77.91.104.22	80
125	spambot malicious download	45942	68.91.226.37	80	172.28.131.186	46722
126	spam bot	45943	172.28.11.150	0	66.200.1.1	25
127	spam bot	45944	172.28.11.150	0	64.180.1.1	25

time-series datasets are available. Many UCI datasets have already been cleaned and are prepared for use.

The dataset collected from different sources is given as input for classification, as shown in Table 3. Due to the presence of compromised servers, few classes are generated.

Performance evaluation metrics

The results of the proposed research will be implemented with some estimated variables, for example: Precision, Sensitivity, Specificity and Accuracy.

The accuracy of a recognition system is measured by correctly identified out of total classified data.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{FN} + \text{TN})} \tag{24}$$

True Positive Rate (TPR) correctly classified data. The FPR measures how often negative samples are incorrectly interpreted as positive due to false positives involving unhealthy samples.

$$\text{TPR} = \frac{\text{TP}}{(\text{TP} + FN)} \tag{25}$$

Precision = number of true positive samples/ (number of true positive samples + number of false negative samples)

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{26}$$

Recall = number of true positive samples / (number of true positive samples + number of false positive samples)

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{27}$$

Where, TP = True Positive TN = True Negative FP = False Positive FN = False Negative.

F-Score: The F-score is an accuracy statistic that combines the precision and recall of a test into a single number. It is used to assess binary categorization systems, which assign examples to one of two classes.

```
spambot malicious download, 46308, 64.222.102.58, 80, 172.28.3.248, 58055
spam bot, 46309, 172.28.3.248, 0, 123.44.92.173, 80
spambot client compromise, 46310, 44.29.203.5, 80, 172.28.130.83, 54407
spambot malicious download, 46311, 64.222.102.58, 80, 172.28.130.83, 44259
spam bot, 46312, 172.28.3.248, 0, 66.200.1.1, 25
spam bot, 46313, 172.28.3.248, 0, 64.180.1.1, 25
spam bot, 46314, 172.28.3.248, 0, 24.145.1.1, 25
spam bot, 46315, 172.28.3.248, 0, 70.98.1.1, 25
spam bot, 46316, 172.28.130.83, 0, 123.44.92.173, 80
                                   precision    recall  f1-score   support

                 compromised_server       1.00      1.00      1.00         2
                               ddos       0.98      1.00      0.99        49
failed attack exploit/iis-asp-overflow       0.00      0.00      0.00         3
                           spam bot       1.00      1.00      1.00        90
          spambot client compromise       0.50      0.87      0.64        31
        spambot malicious download       0.00      0.00      0.00        25

                        avg / total       0.78      0.84      0.80       200

The accuracy score is 84.00%
C:\Users\HP\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:1135:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
```

Fig. 3 Apply SVM classifier

Table 4 Accuracy score of SVM classifier

Topic	Precision	Recall	F1-Score	Support
Compromised server	1	1	1	2
DDoS	0.98	1	0.99	49
failed attack exploit	0	0	0	3
spam bot	1	1	1	90
spambot Client compromise	0.5	0.87	0.64	31
spambot malicious download	0	0	0	25

$$F - score = 2 * (precision * recall)/(precision + recall) \quad (28)$$

SVM classifier implementation

The data are divided into several classes using the SVM classification model, as shown in Fig. 3 and Table 2. In the presence of a compromised server, the classes are classified. This approach provides an accuracy of 84%.

Result output of SVM classifier

Table 4 shows the performance evaluation of a machine learning model used to classify different types of cyber threats in a cloud storage environment. The evaluation metrics used in this table are precision, recall, F1-Score, and support. The model achieved a perfect precision, recall, and F1-Score for the "Compromised server" class, which means that the model correctly identified all instances belonging to this class without any false positives or false negatives. However, the "failed attack exploit" and "spambot malicious download" classes were not detected by the model at all, resulting in a precision, recall, and F1-Score of 0 shown in Fig. 4

Proposed classifier implementation

The data were classified into different groups using the suggested KNN classification [37] with a distance implementation model with altered distance, as illustrated in Fig. 5. Hyperparameters are parameters that are not learned from the training data and must be set before training the model. In SVM and KNN models, adjusting various hyperparameters, such as the regularization parameter (C) and kernel type for SVM, and the number of neighbors (k) for KNN, can enhance model performance. Tuning hyperparameters is a crucial step in building accurate and robust machine learning models. Grid search, random search, and Bayesian optimization are some methods used for optimizing hyperparameters. These methods involve systematically testing different combinations of hyperparameters and evaluating model performance using cross-validation.

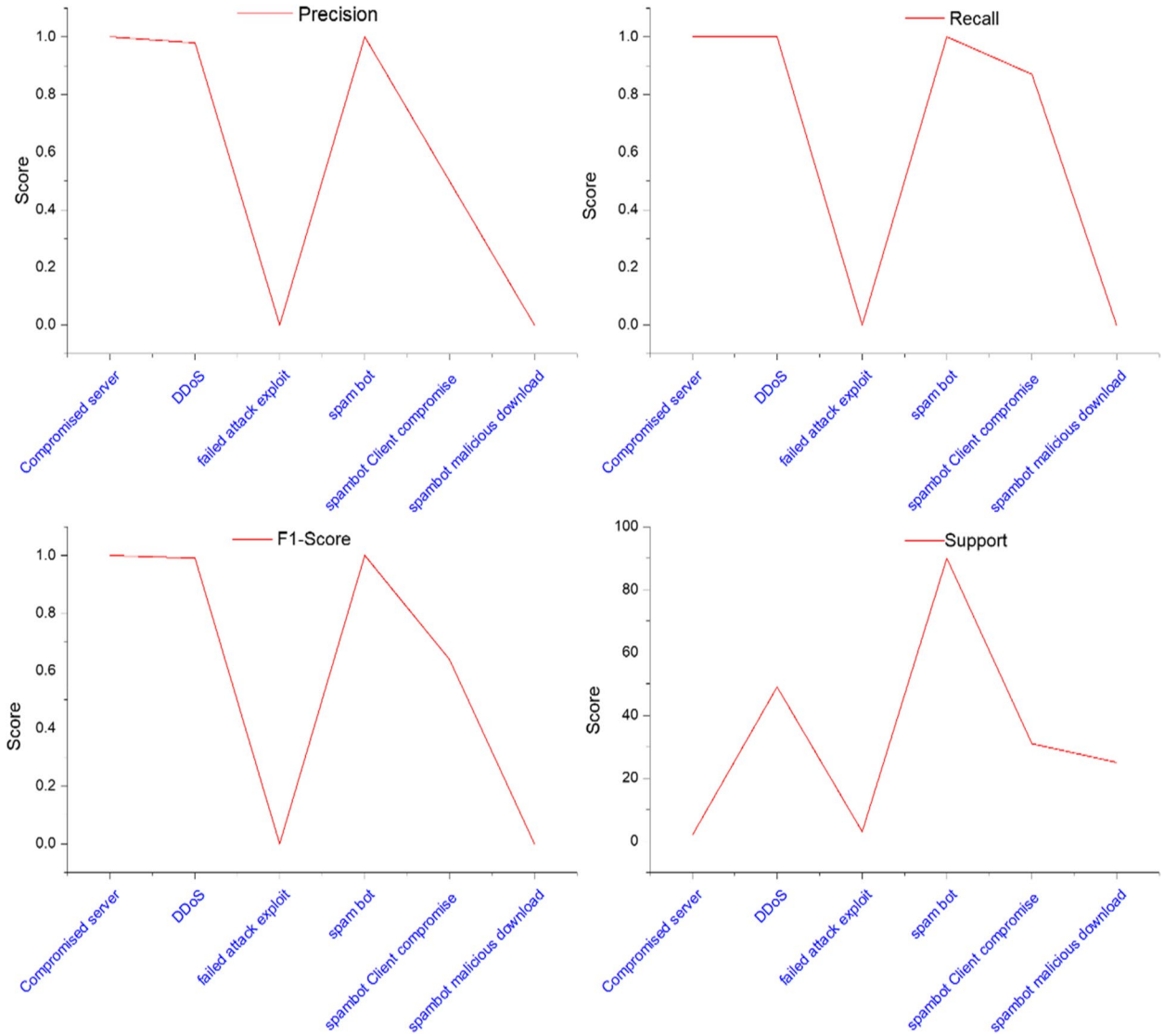

Fig. 4 Analysis of SVM classifier using different parameters

The specific hyperparameters used for SVM and KNN models, as well as any optimization methods employed, would depend on the dataset, project objectives, and available computational resources in the cloud-assisted categorization strategy for secure data storage preservation in smart cities. However, it is important to emphasize that hyperparameter tuning can significantly improve model performance and should be considered a critical stage in model development.

In the presence of a compromised server, the classes are classified. This approach provides an accuracy of 84%.

We have performed the anova test on security parameters of cloud computing in smart cities, which is shown in Table 5.

Result output of the proposed classifier

The accuracy score of the KNN classifier is typically represented as the average of correctly predicted instances (true positives + true negatives) divided by the total number of instances in the dataset. Accuracy of KNN model is 84.0 and SVM is 90.35%. In this paper we have used the cross-validation techniques during the model training phase to estimate the generalization error and evaluate the model's performance. Cross-validation involves partitioning the data set into multiple folds and iteratively training and evaluating the model on different folds. The final performance metric is computed as the average of the performance measures across all the folds.

```
spam bot, 46309, 172.28.3.248, G, 123.44.92.1;3:80
spam bot client compromise, 46310, 44.29.203.5, 80, 172.28.130.83, 54407
spam bot malicious download, 46311, 64.222.102.58, 80, 172.28.130.83, 44259
spam bot, 46312, 172.28.3.248, 0, 66.200.1.1, 25
spam bot, 46313, 172.28.3.248, 0, 64.180.1.1, 25
spam bot, 46314, 172.28.3.248, 0, 24.145.1.1, 25
spam. bot, 46315, 172.28.3.248, 0, 70.98.1.1, 25
spam. bot, 46316, 172.28.130.83, 0, 123.44.92.173, 80
```

	precision	recall	fl-score	support
compromised server	1.00	1.00	1.00	2
ddos	0.98	1.00	0.99	49
failed attack exploit/iis-asp-overflow	0.00	0.00	0.00	3
spam bat	1.00	1.00	1.00	90
spambot client compromise	0.50	0.87	0.64	31
spambot malicious download	0.00	0.00	0.00	25
ay8 / total	0.78	0.84	0.80	200

```
The accuracy score is 84.001 CAUsers\HP\Anaconda3Xlib\site-
packages'sklearn\metricsXclassification.py,1135: UndefinedfttricWarning: Precision and
F-score are ill-defined and being set to 0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
```

Fig. 5 Computation of different parameters using proposed classifier method

Table 5 Anova test on cloud security parameters

Model	Independent Variable		Dependent Variable	T- Test	Standard Deviations	Empirical Evidence
	Beta Test	Std. Error	Beta Test			
(Constant)	0.408	0.313		1.302	0.195	No
Perceived Security	-0.076	0.079	-0.074	-0.958	0.34	No
Service quality	0.273	0.089	0.25	3.078	0.003	Yes
Cost Reduction	0.197	0.079	0.21	2.505	0.014	Yes
IT Background	0.152	0.057	0.182	2.689	0.008	Yes
Perceived Control	0.391	0.08	0.36	4.899	0	Yes

Comparative analysis

Figure 6 and Table 6 present a comparative examination of the capabilities of the SVM and the KNN, respectively. The results of the comparison graph demonstrate that the accuracy level achieved by the KNN classifier is superior to that achieved by the SVM classifier [38].

Figure 7 compares the execution times of the proposed and presented algorithms to demonstrate how they perform. The comparison graph demonstrates that the KNN strategy yields better outcomes than the SVM approach regarding execution time [39].

Figure 8 presents the results of a comparison between the SVM and the KNN in terms of performance. The results of the comparative graph demonstrate that the precision level achieved by the KNN classifier is superior to that achieved by the SVM classifier [40].

A comparative analysis of the performances of SVM and KNN is shown in Fig. 9. P. Su et.al [34] and the author of [41] used the number of abnormal and normal node identification during the data transmission. Thakare et.al [42] and C. H. Wang [43], T. Wang [44] used the behavior of a cluster of received data during the transmission. In the proposed work, we have considered the number of abnormal and normal nodes and cluster behavior during the transmission and the feature matrix of transmitted data. The outcomes of the comparison graph shown in Table 7, the recall level of the KNN classifier is better than the SVM classifier.

Table 5 and Fig. 10 shows the comparative analysis of existing work performed by the different authors with our proposed work in terms of accuracy [46, 47]. The result is that our proposed work outperforms the existing work [48, 49].

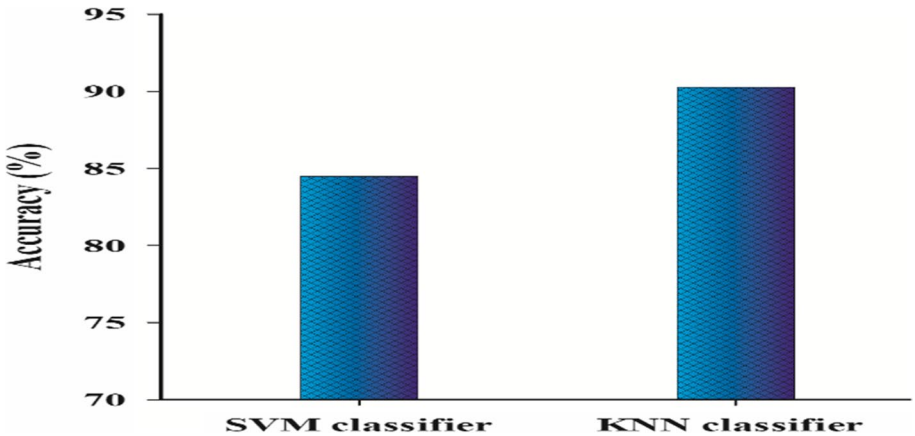

Fig. 6 Accuracy comparison between SVM and KNN

Table 6 Accuracy Score of KNN classifier

Topic	Precision	recall	f1-score	support
Compromised server	0	0	0	2
DDoS	0.45	1	0.62	49
failed attack exploit	0	0	0	3
Spam bot	0.98	1	0.99	90
Spambot Client compromise	0	0	0	25
Spambot malicious download	0	0	0	25

Conclusion and future work

Machine Learning is a powerful method for extracting useful information from a raw dataset. To cluster comparable and dissimilar datasets, the similarity of the input dataset is assessed. In this process, the SVM method is used to classify both comparable and dissimilar data types, and the arithmetic mean of the dataset is calculated to determine the center point. The Euclidean distance is then used to compare the similarity of two data points. Finally, an SVM classifier is employed to classify the clustered data based on the input dataset. This study focuses on the use of the KNN algorithm to predict cardiac disease, where the clustered results are used as input for the classification process. Compared to the current method, the improved technique has higher classification accuracy and shorter execution time. However, the proposed algorithm can be further improved by integrating a hybrid classifier for prediction analysis.

The results of the proposed algorithm were evaluated by comparing it with other existing approaches. However, the study's emphasis on security and privacy has limitations in addressing human-centered aspects that could impede the widespread adoption of smart cities. To enhance public confidence, further research is necessary to visualize the daily experiences of residents living in smart cities and

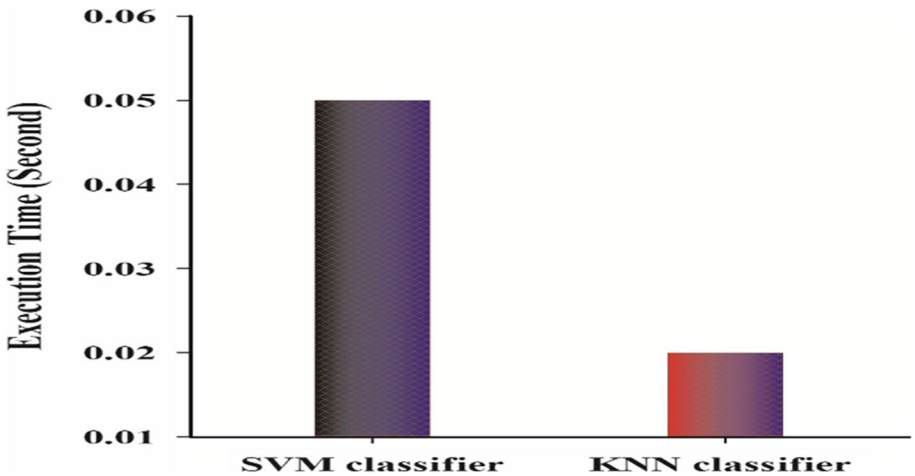

Fig. 7 Execution time comparison between SVM and KNN

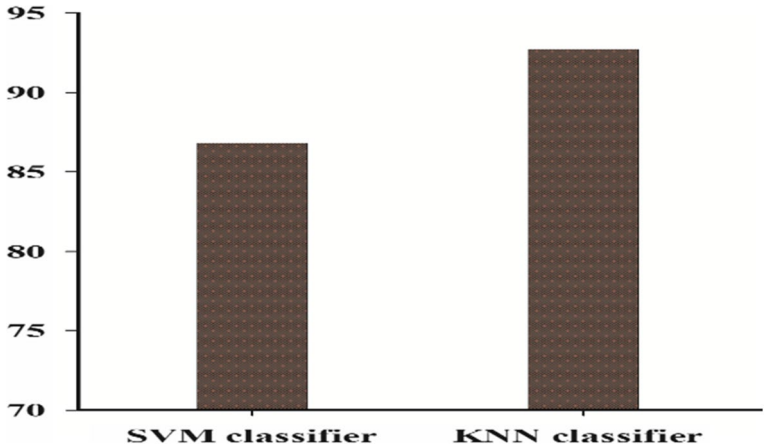

Fig. 8 Precision analysis comparison between SVM and KNN

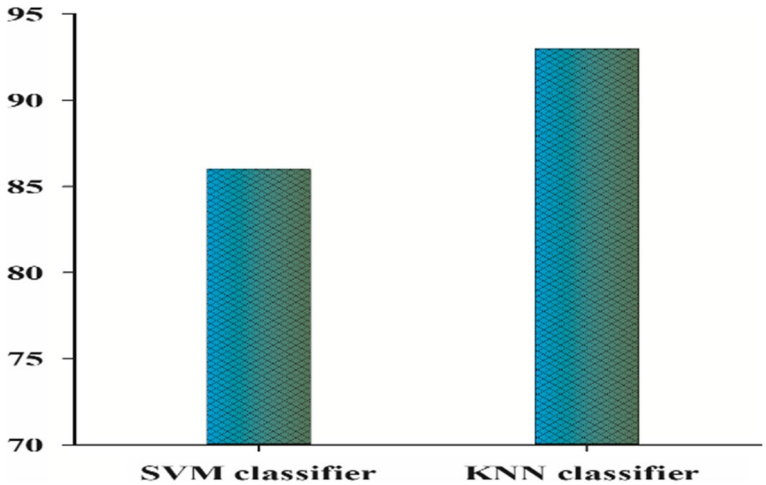

Fig. 9 Recall analysis

Table 7 Related work and comparison of exiting work with accuracy

Author	Year	Method	Accuracy
K. Zhang, X. H. Liang, M. Baura, R. X. Lu, and X. M. Shen [45]	2022	Deep Convolutional Network-based Pipeline	98.40%
V. Roussev and S. McCulley [41]	2021	Ensemble system of Deep Convolutional Neural Network	93.28%
A. Sathya and S. K. S. Raja [42]	2020	Support Vector Machine	93.33%
B. Kakkar, P. Johri, Y. Kumar, H. Park, Y. Son, and J. Shafi [43]	2021	Convolutional Neural Network	95.23%
A. Tembhare, S. S. Chakkaravarthy, D. Sangeetha, V. Vaidehi, and M. V. Rathnam [44]	2022	Lemperl Ziv Markow Algorithm	96.8%
Proposed work	-	Hybrid Method	98.98%

quantify the various interactions and operational difficulties they face. It is important to note that only technological means were considered in this analysis, and the legal and institutional frameworks of a city are equally crucial components that need to be taken into account.

Limitation of proposed methods, the algorithm's performance may be affected by the specific dataset used, and its generalizability to other datasets is uncertain. The proposed algorithm may be further improved by incorporating a hybrid classifier for prediction analysis.

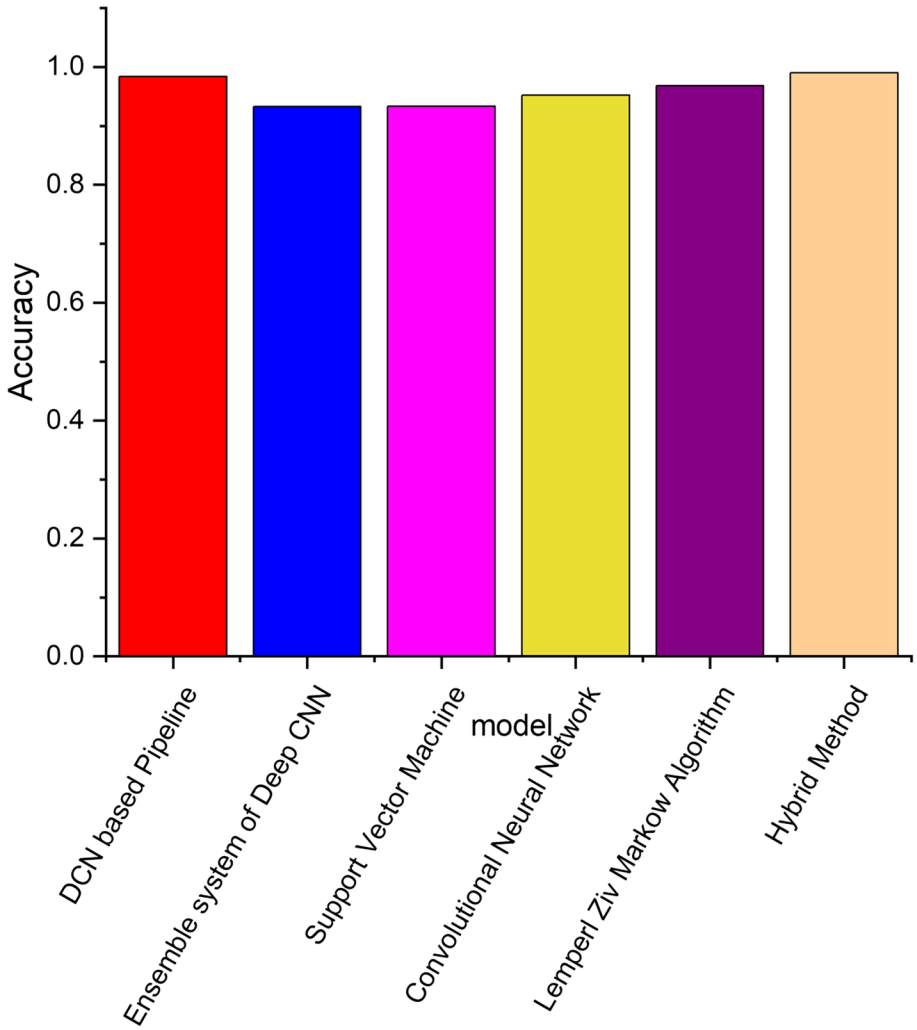

Fig. 10 Comparative analysis of different method with exiting methods

Future work

The use of the law to address trust issues in smart cities is an important topic for future study. Further, smart city projects will benefit immensely from more research aimed at resolving the highlighted obstacles of smart cities (trust challenges, including trust challenges, operational and transition challenges, technology challenges, and sustainability challenges). In future works, we will explore the use of ensemble techniques and compare their performance to the single models used in this study. By using ensemble techniques, researchers could potentially improve the accuracy and reliability of the cloud-assisted categorization strategy and enable more effective data management in smart cities.

Abbreviations

IDS	Intrusion Detection System
DS	Dataset
CAMP	Cloud Application Management for Platforms
DaaS	Desktop as a Service
DRaaS	Disaster Recovery as a Service
SLA	Service-Level Agreement (SLA)
API	Application Programming Interface
SSL	Secure Sockets Layer
VPC	Virtual Private Cloud
VPN	Virtual Private Network
VPS	Virtual Private Server

Acknowledgements

Princess Nourah bint Abdulrahman University Researchers Supporting Project number(PNURSP2023R151), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Authors' contributions

Conceptualization : Ankit Kumar and Surbhi Bhatia ,methodology, Surbhi Bhatia and Saroj Kumar Pandey ,software, Saroj Kumar Pandey and Achyut Shankar ,validation, Achyut Shankar and Carsten Maple formal analysis, Carsten Maple and Arwa Mashat ,investigation, Surbhi Bhatia and Achyut Shankar ,resources, Arwa Mashat and Saroj Kumar Pandey ,data curation, Surbhi Bhatia, writing—original draft preparation, Achyut Shankar ,Carsten Maple, and Arwa Mashat writing- Ankit Kumar, visualization, Areej A. Malibari and Surbhi Bhatia.

Availability of data and materials
The supporting data can be provided on request.

Declarations

Ethics approval and consent to participate
The research has consent for Ethical Approval and Consent to participate.

Consent for publication
Consent has been granted by all authors and there is no conflict.

Competing interests
The authors declare no competing interests.

Received: 27 January 2023 Accepted: 1 June 2023
Published online: 21 June 2023

References

1. Alphonsa MMA, Amudhavalli P (2018) Genetically modified glowworm swarm optimization based privacy preservation in cloud computing for healthcare sector. Evol Intell 11(1–2):101–116. https://doi.org/10.1007/s12065-018-0162-4
2. Anand K, Vijayaraj A, Anand MV (2022) An enhanced bacterial foraging optimization algorithm for secure data storage and privacy-preserving in cloud. Peer Peer Netw Appl 15(4):2007–2020. https://doi.org/10.1007/s12083-022-01322-7
3. Arasi VE, Gandhi KI, Kulothungan K (2022) Auditable attribute-based data access control using blockchain in cloud storage. J Supercomput 78(8):10772–10798. https://doi.org/10.1007/s11227-021-04293-3
4. Balashunmugaraja B, Ganeshbabu TR (2022) Privacy preservation of cloud data in business application enabled by multi-objective red deer-bird swarm algorithm. Knowl Based Syst 236:107748. https://doi.org/10.1016/j.knosys.2021.107748
5. Begum RS, Sugumar R (2019) Novel entropy-based approach for cost-effective privacy preservation of intermediate datasets in cloud. Cluster Comput J Netw Softw Tools Appl 22:S9581–S9588. https://doi.org/10.1007/s10586-017-1238-0
6. Charles VB, Surendran D, SureshKumar A (2022) Heart disease data based privacy preservation using enhanced ElGamal and ResNet classifier. Biomedical Signal Process Control 71:103185. https://doi.org/10.1016/j.bspc.2021.103185
7. Deebak BD, Memon FH, Dev K, Khowaja SA, Qureshi NMF (2022) AI-enabled privacy-preservation phrase with multi-keyword ranked searching for sustainable edge-cloud networks in the era of industrial IoT. Ad Hoc Netw 125:102740. https://doi.org/10.1016/j.adhoc.2021.102740
8. Domingo-Ferrer J, Farras O, Ribes-Gonzalez J, Sanchez D (2019) Privacy-preserving cloud computing on sensitive data: a survey of methods, products and challenges. Comput Commun 140:38–60. https://doi.org/10.1016/j.comcom.2019.04.011
9. Domingo-Ferrer J, Sanchez D, Ricci S, Munoz-Batista M (2020) Outsourcing analyses on privacy-protected multivariate categorical data stored in untrusted clouds. Knowl Inform Syst 62(6):2301–2326. https://doi.org/10.1007/s10115-019-01424-4
10. Zhang J, Peng S, Gao Y, Zhang Z, Hong Q (2023) APMSA: Adversarial Perturbation Against Model Stealing Attacks. IEEE Trans Inform Forensics Secur 18:1667. https://doi.org/10.1109/TIFS.2023.3246766
11. Ebinazer SE, Savarimuthu N, Bhanu SMS (2021) An efficient secure data deduplication method using radix trie with bloom filter (SDD-RT-BF) in cloud environment. Peer Peer Netw Appl 14(4):2443–2451. https://doi.org/10.1007/s12083-020-00989-0
12. Zhou X, Sun K, Wang J, Zhao J, Feng C, Yang Y, Zhou W (2023) Computer vision enabled building digital twin using building information model. IEEE Trans Industr Inf 19(3):2684–2692. https://doi.org/10.1109/TII.2022.3190366
13. Hao JL, Huang C, Ni JB, Rong H, Xian M, Shen XM (2019) Fine-grained data access control with attribute-hiding policy for cloud-based IoT. Comput Netw 153:1–10. https://doi.org/10.1016/j.comnet.2019.02.008
14. Guo Q, Zhong J (2022) The effect of urban innovation performance of smart city construction policies: evaluate by using a multiple period difference-in-differences model. Technol Forec Soc Change 184:122003. https://doi.org/10.1016/j.techfore.2022.122003
15. Abid R, Iwendi C, Javed AR et al (2021) An optimised homomorphic CRT-RSA algorithm for secure and efficient communication. Pers Ubiquit Comput. https://doi.org/10.1007/s00779-021-01607-3
16. Li M, Tian Z, Du X, Yuan X, Shan C, Guizani M (2023) Power normalized cepstral robust features of deep neural networks in a cloud computing data privacy protection scheme. Neurocomputing 518:165–173. https://doi.org/10.1016/j.neucom.2022.11.001
17. Kumar NPH, Prabhudeva S (2021) Layers based optimal privacy preservation of the on-premise data supported by the dual authentication and lightweight on fly encryption in cloud ecosystem. Wirel Pers Commun 121(3):1489–1508. https://doi.org/10.1007/s11277-021-08681-z
18. Dev K, Maddikunta PKR, Gadekallu TR, Bhattacharya S, Hegde P, Singh S (2022) Energy optimization for green communication in IoT using harris hawks optimization. IEEE Trans Green Commun Netw 6(2):685–694. https://doi.org/10.1109/TGCN.2022.3143991
19. Tong D, Chu J, Han Q, Liu X (2022) How land finance drives urban expansion under fiscal pressure: evidence from Chinese cities. Land 11(2):253. https://doi.org/10.3390/land11020253
20. Mishra R, Ramesh D, Edla DR, Mohammad N (2022) Fibonacci tree structure based privacy preserving public auditing for IoT enabled data in cloud environment. Comput Electr Eng 100:107890. https://doi.org/10.1016/j.compeleceng.2022.107890
21. Sun R, Fu L, Cheng Q, Chiang K, Chen W (2023) Resilient pseudorange error prediction and correction for GNSS positioning in urban areas. IEEE Internet Things J 1. https://doi.org/10.1109/JIOT.2023.3235483
22. Dai X, Xiao Z, Jiang H, Alazab M, Lui JCS, Min G, Liu J (2023) Task offloading for cloud-assisted fog computing with dynamic service caching in enterprise management systems. IEEE Trans Industr Inf 19(1):662–672. https://doi.org/10.1109/TII.2022.3186641
23. Narayanan U, Paul V, Joseph S (2022) A novel system architecture for secure authentication and data sharing in cloud enabled big data environment. J King Saud Univ Comp Inform Sci 34(6):3121–3135. https://doi.org/10.1016/j.jksuci.2020.05.005
24. Castiglione A, Pizzolante R, De Santis A, Carpentieri B, Castiglione A, Palmieri F (2015) Cloud-based adaptive compression and secure management services for 3D healthcare data. Future Gener Comput Sys 43–44:120–134. https://doi.org/10.1016/j.future.2014.07.001
25. Dai X, Xiao Z, Jiang H, Alazab M, Lui JCS, Dustdar S, Liu J (2023) Task Co-offloading for D2D-Assisted mobile edge computing in industrial internet of things. IEEE Trans Industr Inf 19(1):480–490. https://doi.org/10.1109/TII.2022.3158974
26. Lian Z, Zeng Q, Wang W, Gadekallu TR, Su C (2022) Blockchain-based two-stage federated learning with non-IID data in IoMT system. IEEE Transactions on Computational Social Systems
27. Rani S, Babbar H, Srivastava G, Gadekallu TR, Dhiman G (2022) Security framework for internet of things based software defined networks using blockchain. IEEE Internet Things J
28. Ning J et al (2020) Dual access control for cloud-based data storage and sharing. IEEE Transactions on Dependable and Secure Computing, Institute of Electrical and Electronics Engineers (IEEE). pp 1–1
29. Sosa-Sosa VJ et al (2022) Improving performance and capacity utilization in cloud storage for content delivery and sharing services. IEEE Transact Cloud Comput 10(1):439–450. Institute of Electrical and Electronics Engineers (IEEE)
30. Yang C et al (2022) Efficient data integrity auditing supporting provable data update for secure cloud storage. Wirel Commun Mobile Comput 2022:1–12 Edited by Junjuan Xia, Hindawi Limited
31. Han Z, Yang Y, Wang W, Zhou L, Gadekallu TR, Alazab M, Gope P, Su C (2023) RSSI map-based trajectory design for UGV against malicious radio source: a reinforcement learning approach. IEEE Trans Intell Transp Syst 24(4):4641–4650. https://doi.org/10.1109/tits.2022.3208245
32. Vijayakumar V, Umadevi K (2021) Protecting user profile based on attribute-based encryption using multilevel access security by restricting unauthorization in the cloud environment. J Ambient Intell Humaniz Comput 12(7):7245–7252. https://doi.org/10.1007/s12652-020-02400-5
33. Javed AR, Ahmed W, Alazab M, Jalil Z, Kifayat K, Gadekallu TR (2022) A comprehensive survey on computer forensics: state-of-the-art, tools, techniques, challenges, and future directions. IEEE Access 10:11065–11089. https://doi.org/10.1109/ACCESS.2022.3142508

34. A, Mary & Sankaralingam, Baghavathi Priya & Mahendran, Rakesh & Gadekallu, Thippa & Ambati, Loknath. Twophase classification: ANN and A-SVM classifiers on motor imagery BCI. Asian J Control. 2022;1(1). https://doi.org/10.1002/asjc.2983.
35. Shabbir A, Shabbir M, Javed AR, Rizwan M, Iwendi C, Chakraborty C (2022) Exploratory data analysis, classification, comparative analysis, case severity detection, and internet of things in COVID-19 telemonitoring for smart hospitals. J Exp Theor Artif Intell. https://doi.org/10.1080/0952813X.2021.1960634
36. Wibowo S et al (2019) Comparing the impact of high pressure, pulsed electric field and thermal pasteurization on quality attributes of cloudy apple juice using targeted and untargeted analyses. Innov Food Sci Emerg Technol 54:64–77. https://doi.org/10.1016/j.ifset.2019.03.004
37. Wu SY, Sun WQ, Ding ZG, Liu SJ (2022) Cloud Evidence tracing system: an integrated forensics investigation system for large-scale public cloud platform. Forensic Sci Int Dig Invest 41:301391. https://doi.org/10.1016/j.fsidi.2022.301391
38. Zhihan LV, Chen D, Haibin LV (2022) Smart city construction and management by digital twins and BIM big data in COVID-19 scenario. ACM Trans Multimedia Comput Commun Appl 18(2s):21. https://doi.org/10.1145/3529395. Article 117
39. Saab S Jr, Saab K, Phoha S, Zhu M, Ray A (2022) A multivariate adaptive gradient algorithm with reduced tuning efforts. Neural Netw 152:499–509
40. Wang H, Gao Q, Li H, Wang H, Yan L, Liu G (2022) A structural evolution-based anomaly detection method for generalized evolving social networks. Comput J 65(5):1189–1199. https://doi.org/10.1093/comjnl/bxaa168
41. Roussev V, McCulley S (2016) Forensic analysis of cloud-native artifacts. Digit Invest 16:S104–S113. https://doi.org/10.1016/j.diin.2016.01.013
42. Sathya A, Raja SKS (2021) Privacy preservation-based access control intelligence for cloud data storage in smart healthcare infrastructure. Wirel Pers Commun 118(4):3595–3614. https://doi.org/10.1007/s11277-021-08278-6
43. Rani S, Babbar H, Srivastava G, Gadekallu TR, Dhiman G (2023) Security Framework for Internet-of-Things-Based Software-Defined Networks Using Blockchain," in IEEE Internet of Things J 10(7):6074–81. https://doi.org/10.1109/JIOT.2022.3223576.
44. Tembhare A, Chakkaravarthy SS, Sangeetha D, Vaidehi V, Rathnam MV (2019) Role-based policy to maintain privacy of patient health records in cloud. J Supercomp 75(9):5866–5881. https://doi.org/10.1007/s11227-019-02887-6
45. Zhang K, Liang XH, Baura M, Lu RX, Shen XM (2014) PHDA: A priority based health data aggregation with privacy preservation for cloud assisted WBANs. Inform Sci 284:130–141. https://doi.org/10.1016/j.ins.2014.06.011
46. Sayour MH, Kozhaya SE, Saab SS (2022) Autonomous robotic manipulation: real-time, deep-learning approach for grasping of unknown objects. J Robot 2585656:14. https://doi.org/10.1155/2022/2585656
47. Saab S Jr, Fu Y, Ray A, Hauser M (2022) A dynamically stabilized recurrent neural network. Neural Process Lett 54(2):1195–1209. https://doi.org/10.1007/s11063-021-10676-7
48. Wen LL et al (2022) A hypothermia-sensitive micelle with controlled release of hydrogen sulfide for protection against anoxia/reoxygenation-induced cardiomyocyte injury. Eur Polym J 175:111325. https://doi.org/10.1016/j.eurpolymj.2022.111325
49. Xu XL et al (2018) An IoT-Oriented data placement method with privacy preservation in cloud environment. J Netw Comput Appl 124:148–157. https://doi.org/10.1016/j.jnca.2018.09.006

Publisher's Note