
Moorthi Kuttiyappan 1, Jothi Prabha Appadurai 2, Balasubramanian Prabhu Kavin 3, Jeeva Selvaraj 4, Hong-Seng Gan 5 and Wen-Cheng Lai 6,*

1 Department of Computer Science and Engineering, Dr.N.G.P. Institute of Technology, Coimbatore 641048, Tamil Nadu, India; moorthicse@gmail.com
2 Department of CSE (Networks), Kakatiya Institute of Technology and Science, Warangal 506015, Telangana, India; ajp.csn@kitsw.ac.in
3 Department of Data Science and Business Systems, College of Engineering and Technology, SRM Institute of Science and Technology, SRM Nagar, Chengalpattu Distric, Kattankulathur 603203, Tamil Nadu, India; ceaserkavin@gmail.com
4 Department of Information Science and Engineering, Jain Deemed to Be University, Global Campus, Bangalore 560069, Karnataka, India; jeeva.selvaraj@jainuniversity.ac.in
5 School of AI and Advanced Computing, XJTLU Entrepreneur College (Taicang), Xi’an Jiaotong-Liverpool University, Suzhou 215400, China; hongseng.gan@xjtlu.edu.cn
6 Department of Electrical Engineering, Ming Chi University of Technology, New Taipei City 24301, Taiwan
* Correspondence: wenlai@mail.mcut.edu.tw or wenlai@mail.ntust.edu.tw

Abstract: One of the industries with the fastest rate of growth is healthcare, and this industry’s enormous amount of data requires extensive cloud storage. The cloud may offer some protection, but there is no assurance that data owners can rely on it for refuge and privacy amenities. Therefore, it is essential to offer security and privacy protection. However, maintaining privacy and security in an untrusted green cloud environment is difficult, so the data owner should have complete data control. A new work, SecPri-BGMPOP (Security and Privacy of BoostGraph Convolutional Network-Pinpointing-Optimization Performance), is suggested that can offer a solution that involves several different steps in order to handle the numerous problems relating to security and protecting privacy. The Boost Graph Convolutional Network Clustering (BGCNC) algorithm, which reduces computational complexity in terms of time and memory measurements, was first applied to the input dataset to begin the clustering process. Second, it was enlarged by employing a piece of the magnifying bit string to generate a safe key; pinpointing-based encryption avoids amplifying leakage even if a rival or attacker decrypts the key or asymmetric encryption. Finally, to determine the accuracy of the method, an optimal key was created using a meta-heuristic algorithmic framework called Hybrid Fragment Horde Bland Lobo Optimisation (HFHBLO). Our proposed method is currently kept in a cloud environment, allowing analytics users to utilise it without risking their privacy or security.

Keywords: big data; security; privacy; Boost Graph Convolutional Network Clustering algorithm; magnify pinpointing based encryption approach; Hybrid Particle Swarm; Grey Wolf Optimization

MSC: 37M10; 68T09

1. Introduction

There are now many electronic medical records, hospital information systems, medical imaging, and other types of data due to the Internet’s quick expansion and the analysis of big data, which has gradually saturated the medical business. Experts in linked sectors predicted the amount of data in the medical industry would be 44 times larger in 2020 than it was in 2009 [1]. Improved design was based on the routing protocol for scalable and secure Internet of Things (IoT)-based healthcare data transmission. A range of wearables and
other Internet of Things devices gather health data. The medical IoT network gathers physiological data, which is then sent to the healthcare big data centre for archival and disease diagnostics. The medical file must be encrypted before transmission in order to preserve patients’ privacy and stop third parties from listening in on private communications. On the protected data, the patient applies an access policy to define the permissible attributes and relationships. Only users possessing the appropriate attribute secret keys—such as a doctor, nurse, anaesthetist, or a patient’s family—are allowed to decrypt the ciphertext. Attribute-based encryption is the name of this encryption technique [2]. Information security and protection have evolved into a fundamental problem that impacts numerous cloud applications. The ease with which cloud administrators can access sensitive data is one of the major concerns with regard to data security and privacy. This concern sharply increases client anxiety and hinders the adoption of distributed computing in many industries, including the financial, governmental, and administrative entities. The security of structured or unstructured data is not addressed by the conventional big data approach. In cases where the risk of disclosing personal information is real, it also has to combine the concepts of insurance and security. The enormous combination of organised and big data necessitates new models for improving safety and security. The growing issue of data security in big data is of great interest to researchers [3].

A patient with unclear symptoms may be diagnosed and treated in many hospitals, and various medical records may be kept, according to the current medical system [4]. As a result, developing a cross-domain, safe data-sharing system is essential to facilitating patient care at various hospitals. For instance, the clinicians at hospital B have access to the examination report created by hospital A. The public clouds are used to store and provide ubiquitous data access for the encrypted medical files produced by various hospitals [5]. The cross-domain access policy for a patient’s protected medical records is determined by the patient. Each member of the medical staff must register with their respective medical facility in order to obtain the attribute secret key needed to decrypt the patient’s encrypted files [6]. Figure 1 explains the role of big data in the healthcare industry [7]. The sources of big data used in healthcare range from wearable technology and search engine server logs to electronic health records. This is a limitless ocean of information that presents an infinite number of opportunities. Knowing how to use this data effectively is crucial. All parties involved in the healthcare system, including healthcare organisations (HCOs), patients, medical professionals, pharmaceutical makers, etc., can benefit from proper storage and analysis instruments. In general, patients’ health improves, doctors can dramatically improve medical results, HCOs can reduce costs and increase operational efficiency, pharmaceutical companies can make better judgements, and other healthcare providers can do so as well.

![Big data in healthcare](image.png)

**Figure 1.** Big data in the healthcare industry.
Unencrypted data transmissions are one of the privacy issues brought on by the volume of data [8,9]. The cloud server, as well as any database server in the cloud, is unreliable. The Febo is a comprehensive health toolbox that helps a person keep track of their bowel movements and discover how their body reacts. [10]. Green computing that uses less energy [11] is used to create the best plans for assigning tasks and [12,13] that addresses energy waste issues in dynamic networking environments. The goal of both of these examples is to cut down on overall energy use. One of our primary areas of focus will be healthcare. Because each person’s genome occupies more than 140 terabytes, healthcare applications require a significant amount of space to safely store genome data [14,15]. As a result of security breaches [16–18], data owners reveal it on big data servers or in the cloud. Password guessing, brute-force attacks, stolen verification attacks, and other security threats target big data storage systems. Users’ and data owners’ confidentiality is not adequately protected by current security measures, which advocate encrypting data before delivering it [19].

As Figure 2 clearly portrays, the privacy assurance of the Big Data in Healthcare system [20], SecPri-BGMPOP, which can offer a solution that incorporates multiple processes, is proposed as a means of addressing the numerous issues associated with privacy and security protection. A brand-new Boost Graph Convolutional Network Clustering (BGCNC) algorithm was initially used to develop effective techniques. We provide a framework to improve BGCNC’s convergence to advance this idea. The advantages we provide in terms of memory and computation are significant. It has the same memory complexity as conventional SGD, and the full gradient descent has the same temporal complexity per epoch. Second, a Magnify Pinpointing-Based Encryption (MPBE) method uses a portion of the identification bit string to create a safe key, preventing identity leakage even if the encryption or encoded data is cracked by an enemy or assault. Using the meta-heuristic algorithmic framework Hybrid Fragment Horde Bland Lobo Optimisation (HPSGWO), an optimal key was then produced. Despite their good performance, traditional algorithms can be improved upon to address their flaws and raise the bar. The traditional PSO method has some drawbacks, such as subpar performance in a number of sectors. The GWO algorithm also has slower convergence, less solution precision, and less efficient local searching, in addition to these drawbacks. The updated place is the primary renovation in the proposed paradigm. Our recommended approach is now maintained in a cloud environment, enabling analytics users to access it without jeopardising their security or privacy.

![Figure 2](image-url)  
**Figure 2.** Big data privacy assurance in the healthcare system.

The contribution of the work is as follows:
- The purpose of the Prvsec-Sanitize is to offer a cloud-based healthcare big data environment with an improved security mechanism.
- The Boost Graph Convolutional Network Clustering (BGCNC) technique is a quick and memory-efficient training algorithm that lowers computational complexity in terms of time and memory measures.
- Even when traditional algorithms perform well, they can still be enhanced to address flaws and increase the bar. The traditional PSO method has certain drawbacks, such
as poorer performance in a number of domains. The GWO algorithm also has poorer solution precision, slower convergence, and less successful local searching in addition to these drawbacks. Thus, more investigation is required to improve integration and robustness.

2. Literature Review

One of the most prominent research fields at the moment is cloud security, and several methods have been put out over the previous few decades. Smart grids make extensive use of software-defined networks (SDN) for communication network management and monitoring. Big data analytics are becoming increasingly significant for SDN-based smart grids. A possible strategy is to examine the vast amounts of data produced by an SDN-based smart grid by utilising machine learning techniques. The optimising and differentially private clustering algorithm (ODPCA) is a technique for selectively private and optimal clustering. For privacy-preserving clustering of mixed data, the ODPCA combines the differentially private K-means algorithm with the K-modes approach. The distribution of the privacy budget is optimised [21] to boost the accuracy of the clustering findings. Using fog computing technology, secure healthcare-sensitive data may be kept in the cloud. A tri-party, one-round authenticated key-agreement mechanism that may generate a session key among participants and enable safe communication can also be represented by bilinear pairing cryptography. Finally, a decoy method might be used to access and safeguard private healthcare data. The suggested method [22] states that when an intruder is discovered accessing system data, fake files are retrieved right away to protect data security.

A blockchain-based system uses secure key management (BC-EKM). In order to build the stake blockchain, a hybrid sensor network is used. The security of key management is compromised by an easily targeted nontrusted base station (BS) due to dynamics. In addition to avianizing the role of BS, the distribution key management schemes also result in a significant and extra expense on sensors. Finally, we carry out in-depth security simulations and analyses [23]. For a number of smart city applications, Holistic Big Data Integrated Artificial Intelligent Modelling (HBDIAIM) has been presented as a way to enhance the security and privacy of the data management interface. To adequately secure the private data management interface in smart city applications, a differential evolutionary algorithm has been incorporated into HBDIAIM. The differential evolutionary algorithm also incorporates the big data analytics-assisted decision privacy approach, which enhances the scalability and accessibility of data in a data management interface based on their corresponding storage location. Additionally, to solve privacy and scalability issues in the data management interface for various smart city applications, the Adaptable Interference Method was created [24].

By recommending a better key management system, this study aims to address the challenges related to the security and privacy concerns of sensitive patient information. The suggested approach also strives to offer a simple and structured key management system. This system has a null rekeying technique and only needs a few key calculations to achieve forward and backward compatibility secrecy. As a consequence, the Healthcare Key Management (HCKM) framework, which strives to decode the same plain text with various keys, is the safe and privacy-preserving key management method for e-health systems. While maintaining an adequate degree of security, HCKM reduces the rekeying overhead for group members and the overhead stated in terms of the number of messages exchanged [25]. A blockchain-based Internet of Things solution for large data transport and privacy protection is facilitated by AI. Using graph modeling, the proposed algorithm first constructs a reliable and scalable system for data collection and transmission. Additionally, the approach based on artificial intelligence is used to extract the subset of nodes and produce efficient healthcare services. Using symmetric-based digital certificates, blockchain allows for secure and private transmission of communication resources [26].

By lengthening the keys in the Data Encryption Standard (DES), the Triple Data Encryption Standard (TDES) technique offers a way that is significantly easier to use while
securing data privacy and preventing assaults. The experiment’s findings demonstrate that the TDES technique is effective in securing and safeguarding substantial volumes of healthcare data stored in the cloud [27]. This research presents a unique encryption technique using Serpent, an Advanced Encryption Standard (AES), and elliptic curve cryptography to protect healthcare data in IoT-enabled healthcare infrastructure. The suggested hybrid encryption approach improves healthcare data security by integrating symmetric and asymmetric-based encryption techniques. This technique additionally makes use of an elliptic curve-based digital signature to ensure the data’s integrity [28].

The Internet of medical things (IoMT) network receives massive amounts of big data, also known as health data, and registers a sizable number of patients and devices each day. To prevent misuse, this patient data should remain private and secure on the IoMT network. To achieve such data privacy and security, the interplanetary file system (IPFS) and a three-level/tier network have been proposed [29].

An enhanced architecture for safe and scalable IoT-based healthcare data transfer is based on the routing protocol. First, a variety of Internet of Things (IoT) devices, including wearables and sensors, collect health data. Utilizing data reduction and cleaning procedures, the raw data are preprocessed. Principal Component Analysis (PCA) and K-Nearest Neighbour (KNN) imputation are used to minimize the dimensionality of the data. The preprocessed data is used to extract the features using modified local binary patterns (MLBP). The FDT-RPL protocol for low-power and lossy networks enhances overall data transmission security by integrating the Butter Ant Optimization (BAO) algorithm with the fuzzy dynamic trust-based RPL algorithm [30]. None of these methods, therefore, will guarantee knowledge protection, which is necessary for both privacy and usefulness. To safeguard medical records, an effective anonymization approach is nevertheless required. Recent developments have shown that sensitive data, including proprietary knowledge, is kept, frequently by meta-heuristic algorithms. These methods are designed to generate an ideal key for the sanitization process. It is demonstrated that these algorithms produce superior results when compared to traditional methods. When compared to security, privacy is valued as a benefit, which is why it is important for both purchasing and selling in the business. Therefore, it is critical to strike a balance between security and privacy. A number of research frequently employed query and k-anonymity to address privacy concerns. However, these methods need a lot of computational resources and time. Our recommended approach offers a more reliable security system for cloud-based large data in healthcare.

The remaining parts of the paper are laid out as follows: The proposed work is discussed in Section 3, Results and Discussions are presented in Section 4, and the conclusion and the work to come are discussed in Section 5.

3. Proposed Method

The SecPri-BGMPOP approach is suggested and can offer a solution that involves several different steps in order to handle the numerous problems relating to security and protecting privacy. The Boost Graph Convolutional Network Clustering (BGCNC) algorithm, which reduces computational complexity in terms of time and memory measurements, is first applied to the input dataset to begin the clustering process. Figure 3 shows the architecture of the data security and privacy model. The way Boost Graph convolutional network clustering operates is as follows: the BGCNC training algorithm is fast and memory-efficient. It samples a block of nodes linked to a dense subnetwork at each phase and restricts the neighbourhood search towards this substring. The dense subgraph is identified via a graph-clustering algorithm. Memory and computational efficiency are greatly increased by using this straightforward yet efficient technique. An encryption method based on Magnify Pinpointing prevents identity leaks even if an adversary or attacker decodes the key or encrypted material by creating a safe key utilising a portion of the identification bit string. Based on the Hybrid Fragment Horde Bland Lobo Optimisation framework, an enhanced meta-heuristic algorithmic framework, an optimum key was
obtained. Our suggested SecPri-BGMPOP is intended to offer a better security method for cloud-based big data in healthcare.

Figure 3. The main architecture of the data security and privacy model.

3.1. Graph Convolutional Network Clustering (GCNC)

The best of both worlds is achieved by the graph convolutional network clustering technique; this has the same memory complexity as conventional SGD and full gradient descent has the same temporal complexity per epoch [31]. Consider the scenario in which each batch’s embeddings are determined for a collection of nodes B from layer 1 to layer L. Given that every layer of computation uses the same subgraph $A_{B,B}(\text{Links within } B)$, we may assume the number of edges in this batch, $\|A_{B,B}\|_0$. To indicate the embedding utilisation, we need build a batch B that maximises within-batch edges in order to increase embedding utilisation. In order to link the effectiveness of SGD updates with graph clustering techniques, Figure 4 shows the complete graph $G$ and the graph with the clustering partition $G$ as examples of the community expansion. Cluster-GCN, as can be shown, can concentrate on the neighbours within each cluster rather than conducting a thorough neighbourhood search.

A graph $G = (V, \varepsilon, A)$ consists of $N = |V|$ vertices and $|\varepsilon|$ edges such that an edge among any two vertices $i$ and $j$ represents their similarity. The corresponding adjacency matrix $A$ is an $N \times N$ sparse matrix with $(i, j)$ entry equaling 1 if there is an edge between $i$ and $j$ and 0 otherwise. $G$’s partition its nodes into $c$ groups: $V = \{V_1, \ldots, V_c\}$, where $V_i$ consists of the nodes in the $t$-th partition, $L$ is the number of layers, $F$ is the number of features, $N$ is the number of nodes, and $b$ is the batch size. As a result, we have $c$ subgraphs

$$\mathcal{G} = [G_1, \ldots, G_c] = \{\{v_1, \varepsilon_1\}, \ldots, \{v_c, \varepsilon_c\}\}$$ (1)

where each $\varepsilon_i$ is made up only of the connections connecting the nodes in $v_i$. The adjacency matrix is divided into $c^2$ submatrices after nodes are reorganised as

$$A = \mathcal{A} + \Delta = \begin{bmatrix} A_{11} & 1 & A_{1c} \\ 1 & 0 & 1 \\ A_{c1} & 1 & A_{cc} \end{bmatrix}$$ (2)

and

$$\mathcal{A} = \begin{bmatrix} A_{11} & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & A_{cc} \end{bmatrix}, \Delta = \begin{bmatrix} 0 & 1 & A_{1c} \\ 1 & 0 & 1 \\ A_{c1} & 1 & 0 \end{bmatrix}$$ (3)
Figure 4. The distinction between our suggested cluster technique and conventional graph convolution is the neighbourhood expansion. The extension of neighbourhood nodes begins at the red node. The growing convolution suffers from hyperbolic neighbourhood growth; however, our technique can stop expensive neighbourhood expansion.

The links within \( G_t \) are contained in each diagonal block \( A_H \) is a \([v] \times [v] \) adjacency matrix. The adjacency matrix for the graph \( G_t \), \( \bar{A} \) is the adjacency matrix for graph \( G \); \( A_H \) contains the connections between its two partitions \( v \), \( \Delta \). Similar to this, we can divide the \( \hat{X} \) and \( Y \) into \([v] \) as \([X_t] \) according to the partition \([Y_t] \) where \( X_t \) and \( Y_t \) are made up.

The calculation allows for the decomposition of the objective function of GCN into many groups (clusters). The final embedding matrix is if we use \( \bar{A}^t \) to represent the normalised version of \( \bar{A} \).

\[
Z^{(L)} = \bar{A}^t \sigma \left( \bar{A}^t \sigma \left( \ldots \sigma \left( \bar{A}^t X W^{(0)} \right) W^{(1)} \right) \ldots \right) W^{(L-1)}
\]  

(4)

This is significantly less difficult to construct than the neighbourhood search method utilised in earlier SGD-based training methods because \( \bar{A} \) has a block-diagonal structure (remember that \( \bar{A}_H \) is the matching diagonal block of \( \bar{A} \)). Additionally, it has broken down into

\[
L_{\bar{A}} = \sum_i |v_i| L_{\bar{A}_H} \quad \text{and} \quad L_{\bar{A}_H} = \frac{1}{|v_i|} \sum_{i \in v_i} \text{loss}(y_i, z_i^{(L)})
\]  

(5)

The decomposition form in (3) and (4) serves as the basis for the Cluster-GCN (5). At each step, we sample a cluster \( v_i \) and the conduct SGD to update based on the gradient of \( L_{\bar{A}_H} \) and this only requires the subgraph \( A_H \), the \( X_t \), and the \( Y_t \) on the current batch models \( \{w\} \) . Compared to the neighbourhood search strategy employed in earlier SGD-based training methods, the implementation merely calls for matrix components’ forward and retrograde transmission (one block of (5)). As previously mentioned, the embedding utilisation is similar to the within-cluster linkages for each; therefore, these are exactly what we require. (1) Assuming that each node and its neighbours are often found in the same cluster, neighbourhood nodes have a high probability of remaining in the same cluster after a few hops. (2) We need to design a partition to reduce the number of between-cluster links since we replace \( A \) with its block diagonal approximation \( A \) and the error is proportional to those links.

Each node in \( v_i \) of \( A_H \) only links to other nodes inside of \( v_i \). Each batch’s computation will only involve matrix products \( \bar{A}_H X^{(l)} w^{(l)} \) and a few element-wise operations; thus the
total computation time is $O(\|A\|_0 F + bF^2)$. As a result, $O(\|A\|_0 F + NF^2)$ is the overall temporal complexity per epoch. On average, each batch only needs to compute $O(bL)$ embeddings, which is exponential to $L$ rather than linear. For embedding storage, we simply must load $b$ examples per batch, and the per layer’s extracted features are saved using $O(blF)$ memory.

Boost Graph Convolutional Network Clustering (BGCNC)

To achieve good computational and memory complexity, two potential concerns still exist:

- Certain links (the part in $\Delta$ Equation (2)) are eliminated from the graph after it has been partitioned. As a result, the performance can be impacted.
- Algorithms for graph clustering frequently group related nodes together. As a result, a cluster’s distribution may deviate from the original data set, which could cause a skewed estimation of the complete gradient while doing SGD updates.

Based on the label distribution of each cluster, we determine its entropy value. When compared to random partitioning, it is evident that most clusters have lower entropies, which suggests that the label distributions of the clusters are skewed towards particular labels. This raises the variance between batches and could have an impact on SGD convergence [32].

We suggest a stochastic multiple clustering strategy to incorporate between-cluster linkages and lessen volatility across batches in order to overcome the aforementioned problems. Then, we divide the graph into $p$ clusters $V_1, \ldots, V_p$ using a sizable $p$. Instead of taking into account just one cluster while building a batch $B$ for an SGD update, we pick $q$ clusters at random, denoted as $t_1, \ldots, t_q$ and add their nodes as part of the batch using the notation \( \hat{V} = \cup t_1 \cup \ldots \cup t_q \). Moreover, the connections between the selected clusters are re-added.

\[
\{A_{ij}|i,j \in t_1, \ldots, t_q\} \quad (6)
\]

In this way, the cluster combinations reduce the variance between batches and re-integrate the between-cluster linkages. Algorithm 1 presents our final Boost Graph Convolutional Network Clustering Technique.

Algorithm 1: Boost Graph Convolutional Network Clustering

1: **Input:** Graph A, Feature X, Label Y
2: **Output:** Node representation $\hat{X}$
3: Partition graph nodes into $c$ clusters $V_1, V_2, \ldots, V_c$
4: For $\text{iter} = 1, \ldots, \text{max iter}$ do
5: Randomly choose $q$ clusters, $t_1 \ldots t_q$ from $V$ without replacement;
6: From the subgraph $\hat{G}$ with nodes $\hat{V} = \bigcup t_1 \cup \ldots \cup t_q$ and links $A_{\hat{V}, \hat{V}}$;
7: Compute $g \leftarrow \nabla L_{A_{\hat{V}, \hat{V}}}$ (loss on the subgraph $A_{\hat{V}, \hat{V}}$);
8: Conduct adam update using gradient estimator $g$
9: **Output:** $\{W_l\}_{l=1}^L$

The data is divided into groups using the Boost Graph Convolutional Network Clustering technique, and each group is then handed to a faster parallel process to increase the computational complexity in terms of time and memory measurements. The magnify pinpointing-based encryption method is then applied to the data groups, which transforms the data into a different format based on the key value generated. To save time, the encryption, decryption, and key creation processes all use the magnify pinpointing-based approach.

3.2. Magnify Pinpointing-Based Encryption (MPBE)

Only key creation, encryption, and decryption times are calculated in the suggested method. Currently, network overhead is a problem. This framework shows how user 1 and user 2 can access each other’s data by using magnify-pointing-based encryption as
Role-based classification is faster with identity-based encryption [33]. With these techniques, users are given access to the data after authenticating their identities. This plan is initially put into practice on proxy servers to remove unwanted users. The identities of allowed users are recorded on proxy servers, and whenever an unauthorized user tries to access the server’s service, their access is revoked because no key matches that identity. As a result, in order to access the service, each user must register their identification. Security is becoming increasingly in demand as a result of technological advancements like cloud computing. This system is used by several security companies to protect user data and information [34]. Uploading and downloading of data or files is a crucial component of cloud security implementation. The effectiveness of an IBE scheme is also significantly influenced by network congestion and service speed. The Algorithm 2 describes the key generation process. However, in order to improve time efficiency, the suggested method described in Section 3.2 focuses primarily on key creation, encryption, and decryption. The only times that are calculated in the proposed method are those for key generation, encryption, and decryption. At this time, network overhead is compromised. Using IBE as a security, authentication, and authorization tool, this system illustrates how Bob and Alice can access each other’s data. An authorized user’s identity cannot be obtained by an unauthorized user because PKG stores user identities in Lagrange polynomial equation form, which is computationally challenging for an outsider to decipher and obtain the user’s original identity. Because they communicate these identities using secure socket connections (SSL), when Bob sends his identification to Alice in the middle of the network, an attacker cannot obtain the identity. Hash-based key generation techniques are computationally costly; hence, pairing-based key generation techniques are used instead.

As mentioned, the demand for security is rising because of technological advancements like cloud computing. This system is used by several security companies to protect user data and information. Uploading and downloading of data or files is a crucial component of cloud security implementation.

Let $G$ stand for a collection of prime order $p$. Group $G$ forms an accurate bilinear map onto Group $G_1$. $G_1$’s bilinear map representation is given by the formula $e: G \times G_1 \rightarrow G_2$.
where \( g \) is the group \( G \)'s generator. The group size is determined by a security parameter, and each identity is represented by four strings, each of length \( n \).

\[
SL = (sl_1, sl_2, sl_3, \ldots, sl_n).
\] (7)

The \( SL \) bit strings length is utilised to produce fixed length \( n \) bit strings. The suggested magnify pinpointing-based encryption algorithm includes the following phases.

**Algorithm 2: Key generation**

1. Private key generator (PKG) starts the setup process and decides the security parameters like the level of bits and type of curves.
2. Bob obtains the master public key from PKG.
3. Bob authenticate himself by issuing his identity to PKG and receives the private key for encrypting the data.
4. Similarly, Alice obtains the master public key from PKG.
5. Alice authenticate herself by issuing his identity to PKG and receives the private key for encrypting the data.
6. Bob sends his identity to Alice for generating the public key related to Bob’s identity. Alice will use this key to decrypt the data authenticated by Bob.
7. Alice obtains Bob’s data from the database and decrypts it for accessing.

3.2.1. Initial Phase

The system parameters are created first. From \( Z_p \) an underground is chosen at random. An accidental maker \( g \) from the set \( G \) is selected such that \( g \in G \), the value \( g^I = g^\alpha \) is fixed, and \( g^2 \) is randomly chosen from the set \( G \). A chance quantity \( u \) is selected such that \( u' \in G \) and a random \( n \)-length vector such that \( U = \{u_i\} \) after selecting all authority parameters. Finally, the public parameters \( g \), \( g^1 \), \( g^2 \), \( u' \), and \( U \) are broadcast along with the master key \( g^\alpha^2 \).

3.2.2. Generation Phase

Let \( v \) be the \( n \)-bit string \( SL \) for the user, and let \( V \subseteq \{1, \ldots, n\} \) be the collection of all \( i \) for which \( v_i = 1 \). \( V \) is split into two different, namely \( V_1 = \{v_1, v_2, \ldots, v_m\} \) and \( \{V_{r_1}, V_{r_2}, \ldots, v_{m}\} \) such that \( m + r_m = n \), where \( V_{r_1} \) stands for a random value that is introduced to the suggested approach to increase security. The private key corresponding to identity \( v \) is obtained by selecting a random value, as indicated in Equation (8). \( u' \prod_{i \in V} v_i \) is the formula for group operation during the key generation:

\[
d_v = \left( u' \prod_{i \in V} v_i \right)
\] (8)

\( U = \{u_1, u_2, \ldots, u_n\} \) and \( V = \{v_1, v_2, \ldots, v_m\} \) such that \( m < n \). Now, a polynomial function using the Lagrange coefficient method is created and we perform polynomial interpolation. We can conceal \( v \), which can be effectively reconstructed from the available data points with the aid of polynomial interpolation. For the suggested strategy, the polynomial equation and Lagrange coefficient is

\[
\Delta_{i,v}(x) = \sum_{i=0,k \in V}^n \left( \prod_{0<i<n, j \notin i} \frac{x - x_j}{x_i - x_j} \right)y_k
\] (9)

where \( x = u_i \) and \( y = v_k \).

Every user identity’s random set \( u_i \) is generated once, and each identity’s Lagrange coefficient is generated using the same \( u_i \) value. M-terms of identity will be used by authority value as a result of which a challenger will never learn the authentic user’s original identity. As a result, it will be challenging to retrieve or deduce the key created for
a specific $SL$. What if every user identity and $U$ value are identical? The challenger will be unable to infer anything from the key in such case since $\Delta_{uv}(x)$ produces an error of zero. This situation is unique. The greatest error between any two subsequent nodes will be shown, and the error created will be zero.

3.2.3. Encryption Phase

Let message $M (M \in G_1)$ and “c” be a random number selected from $Z_p$. Equation can be used for encryption for some identities $v(10)$, Clusters 1, 2, and 4. Encryption using three-key TDEA is prohibited unless otherwise approved by another NIST guidance.

$$C = \left( e \left( g_1, g_2 \right)^c M, g_c, \left( u' \prod_{i \in V} V_i \right)^c \right)$$  \hspace{1cm} (10)

3.2.4. Decryption Phase

Let $C = (C_1, C_2, C_3)$ be a valid cypher text under user identity $v$ for message $M$. Then, using cypher text $C$ as a key, $d_v = (d_1, d_2)$ as given in Equations (11)–(13):

$$C_1 e^{(d_1, C_3)} = \left( e \left( g_1, g_2 \right)^c M \right) \frac{e^{g_2^\prime \left( u' \prod_{i \in V} v_i \right)^c}}{e^{g_2 \frac{1}{\left( u' \prod_{i \in V} v_i \right)^c}}},$$ \hspace{1cm} (11)

$$= \left( e \left( g_1, g_2 \right)^c M \right) \frac{e^{g_2 \left( u' \prod_{i \in V} v_i \right)^c}}{e^{g_2 \left( u' \prod_{i \in V} v_i \right)^c}} \frac{e^{g_2^\prime \left( u' \prod_{i \in V} v_i \right)^c}}{e^{g_2^\prime \left( u' \prod_{i \in V} v_i \right)^c}},$$ \hspace{1cm} (12)

$$= M$$ \hspace{1cm} (13)

Next the generated key is optimised using the Hybrid Fragment Horde Bland Lobo Optimization algorithm.

3.3. Hybrid Fragment Horde Bland Lobo Optimization

3.3.1. Traditional PSO Algorithm

Three vectors are used in the computations of the PSO algorithm. They are x-vectors, p-vectors, and v-vectors, accordingly [35]. The p-vector (pbest) designates the place where the particle has found the object, and the x-vector tracks the particle’s current location inside the search zone (best response yet). Particle velocity, which forecasts where each succeeding particle will move throughout the course of that iteration, is likewise included in the v-vector. Initial random displacement of the particles occurs in preset directions. Gradually altering the particle’s orientation allowed it to start moving on its own in the direction of the prior best position. Then it searches the area for the optimal spots to perform some fitness-related tasks, using the formula $fit = S^m - S$. Here, the particle’s position is specified as $M \in s^m$, although its speed is given as $\vec{w}$. These two variables are initially chosen at random, and they are subsequently updated repeatedly using the two formulas in Equation (14):

$$\vec{w} = \omega \vec{w} + c_1 r_1 (\vec{p} - \vec{M}) + c_2 r_2 (\vec{f} - \vec{M})$$ \hspace{1cm} (14)

The inertia weight, or $\omega$ in this instance, is a user-defined behavioural parameter that controls the degree of particle velocity recurrence. The particles implicitly interact with one another since the even-before position of the particle (pbest position) is $\vec{q}$, and the prior-best location of the particle inside the swarm (gbest position) is $\vec{f}$. The acceleration constants are $c_1, c_2$. Utilizing the stochastic variables, this is graded, $r_1, r_2 \sim U(0, 1)$. The equation shows that the particle is moved to the next site in the search zone by adding the velocity to its present position (15), as shown in Figure 6, regardless of fitness gains [36].

$$\vec{M} \leftarrow \vec{M} + \vec{w}$$ \hspace{1cm} (15)
Hierarchical search agents exist in the GWO algorithm [37]. Equations (16) and (17) use mathematics to illustrate how encircling occurs as grey wolves hunt their prey (17):

$$
\vec{B} = \left| \vec{E} \cdot \vec{M}_q(u) - \vec{M}(u) \right|
$$ (16)

$$
\vec{M}(u + 1) = \vec{M}_q(u) - H \cdot \vec{B}
$$ (17)

where $u$ the current iteration is handed to the term “coefficient vectors” and is used to describe $H$ and $\vec{E}$. Grey wolves have a special ability to locate their prey and surround it. Using the heightened awareness of potential prey sites of alpha, beta, and delta wolves, these grey wolf hunting behaviours are statistically replicated. Regardless of whether further solutions are necessary, the top three are taken into account. Below are the mathematical Equations (18)–(24):

$$
\vec{B}_\alpha = \left| \vec{E}_1 \cdot \vec{M}_\alpha - \vec{M}(u) \right|
$$ (18)

$$
\vec{B}_\beta = \left| \vec{E}_2 \cdot \vec{M}_\beta - \vec{M}(u) \right|
$$ (19)

$$
\vec{B}_\delta = \left| \vec{E}_3 \cdot \vec{M}_\delta - \vec{M}(u) \right|
$$ (20)

$$
\vec{M}_1 = \vec{M}_\alpha - H_1 \cdot \left( \vec{B}_\alpha \right)
$$ (21)

$$
\vec{M}_2 = \vec{M}_\beta - H_2 \cdot \left( \vec{B}_\beta \right)
$$ (22)

$$
\vec{M}_3 = \vec{M}_\delta - H_3 \cdot \left( \vec{B}_\delta \right)
$$ (23)

$$
\vec{M}(u + 1) = \frac{\vec{M}_1 + \vec{M}_2 + \vec{M}_3}{3}
$$ (24)

Despite their good performance, traditional algorithms can be improved upon to address their flaws and raise the bar.

3.3.3. Hybrid Fragment Horde Bland Lobo Optimization

As mentioned, traditional algorithms can be improved upon despite having good performance to solve the shortcomings and raise the bar. There are some flaws in the conventional PSO algorithm, including inferior performance across a variety of fields. Along with these disadvantages, the GWO algorithm also has slower convergence, worse solution preci-
sion, and less effective local searching. Therefore, additional research is needed to enhance robustness and integration. These issues are dealt with in this work using a brand-new hybrid methodology. In the suggested Hybrid Fragment Horde Bland Lobo Optimization, the PSO method’s criteria are blended with the GWO algorithm. Equations (15) and (16) illustrate the mathematical model of the prey enclosure in the proposed approach, whereas Equation (17) shows the hunting strategy’s mathematical model (18). The location has been updated, which is the main reformation in the recommended paradigm. Equation (19) illustrates the updating of the position in our Hybrid Fragment Horde Bland Lobo Optimization model, where \( \vec{M} \) denotes the velocity for updating the location of PSO, as indicated in Equations (18) and (24):

\[
M(u + 1) = \frac{\vec{M}_1 + \vec{M}_2 + \vec{M}_3 + \vec{M}}{4}
\]  

(25)

Again, in the classic PSO method, \( c_1 \) and \( c_2 \) are regarded as acceleration constants; however, in the recommended Hybrid Fragment Horde Bland Lobo Optimization model, \( c_1 \) and \( c_2 \) fluctuate in accordance with the values 0.1, 0.3, 0.5, 0.7, and 1. The optimal key is generated using the Hybrid Fragment Horde Bland Lobo Optimization. Algorithm 3 presents the ideal key selection based on Hybrid Fragment Horde Bland Lobo Optimization.

**Algorithm 3: Hybrid Fragment Horde Bland Lobo Optimization**

1: \( M_j \) is the grey wolf population where \( j = 1, 2, N \). Here, \( M_a, M_b \) and \( M_d \) denote the best searching agent, the second-best searching agent, and the third-best searching agent, respectively. Moreover, \( e \) is the components, and \( H, E \) are coefficients. The goal of this algorithm is to output the best searching agent, \( M_a \).
2: 
3: Set initial values to the \( M_j \)
4: Set initial values to \( e, H, \) and \( E \) also
5: Measure the fitness values of each searching agent, \( M_a, M_b, \) and \( M_d \).
6: while (\( u < \max \)) do
7:   
8:   for each searching agent, do
9:     
10:     Revise the present location of the searching agents using Equation (25)
11:   }
12: Revise \( e, H, \) and \( E \)
13: Assess fitness values for all searching agents
14: Revise \( M_a, M_b, \) and \( M_d \)
15: \( u := u + 1 \)
16: }
17: return \( M_a \)
18: }

4. Results and Discussion

Java code for clustering and the SecPri-BGMPOP technique were used in the trials, which were done under Windows utilising HADOOP jars in CloudSim. Python was used in the development of the suggested procedure. Python was used in the development of the suggested procedure. The Universiti Kebangsaan Malaysia faculty of education provided the autism datasets. The autism datasets used in this study were gathered from autistic children in various age groups. These include the 24-month dataset for autistic children, which contains 26 attributes and 209 instances; the 30-month dataset for autistic children, which contains 29 attributes and 209 instances; the 36-month dataset for autistic children, which includes 31 attributes and 234 instances; and the 48-month dataset for autistic children, which includes 33 attributes and 302 instances. There are three score alternatives for each dataset, which are autism diagnostic data: \( z = 0, v = 5, \) and \( x = 10 \). The cut-off numbers varied for each type of dataset at 71, 95, 100, and 105, respectively.
Our research’s experimental findings are contrasted with the current system. By tracking the amount of time and memory used during execution, the SecPri-BGMPOP technique is assessed. The whole SecPri-BGMPOP execution duration in CloudSim is precisely calculated and measured in milliseconds (ms). The memory utilisation was effectively organised using the Java-based CloudSim programme. Kilobytes are used to measure the size of the SecPri-memory BGMPOP’s footprint and the stack memory needed for newly generated objects (kb).

The Boost Graph Convolutional Network Clustering approach’s SecPri-BGMPOP threshold is “1” for time and memory efficiency tests. Table 1 lists the results for various cluster sizes. Then, in terms of time and memory requirements, the proposed system is compared with FCM [38] and WFCM [39] and the Boost Graph Convolutional Network Clustering employed in previous work.

Table 1. Cluster-Based Time, Memory Measures for SecPri-BGMPOP Threshold Value “1”.

<table>
<thead>
<tr>
<th>Number of Clusters</th>
<th>Time (s)</th>
<th>Memory (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31.569</td>
<td>1520</td>
</tr>
<tr>
<td>2</td>
<td>30.638</td>
<td>1613</td>
</tr>
<tr>
<td>4</td>
<td>30.456</td>
<td>1663</td>
</tr>
</tbody>
</table>

According to a graph drawn for time against cluster count (Figures 7 and 8), the process takes less time to complete the more clusters there are. Because many mappers can run simultaneously, the procedure can be completed more quickly plotted as shown in Figures 7 and 8. It is discovered that the least amount of memory is needed to complete a process when the number of clusters is low because higher cluster counts also result in higher memory allocation.

![Figure 7](image1.png)  
**Figure 7.** Clustering-based time measure for SecPri-BGMPOP threshold value “1”.

![Figure 8](image2.png)  
**Figure 8.** Cluster-based memory measures for SecPri-BGMPOP threshold value “1”.

Table 2 lists the clustering-based time and memory metrics for the “2” SecPri-BGMPOP threshold, and Figures 9 and 10 display graphs of these measures. The findings demonstrate that when the number of clusters rises, both time and memory are decreased since the threshold chosen is the best option.

Table 2. Clustering-Based Time and Memory Measures for SecPri-BGMPOP Threshold Value “2”.

<table>
<thead>
<tr>
<th>Number of Clusters</th>
<th>Time (s)</th>
<th>Memory (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>29.55</td>
<td>1325</td>
</tr>
<tr>
<td>2</td>
<td>27.55</td>
<td>1265</td>
</tr>
<tr>
<td>4</td>
<td>26.45</td>
<td>1232</td>
</tr>
</tbody>
</table>

Figure 9. Clustering-based time measure for SecPri-BGMPOP threshold “2”.

Figure 10. Clustering-based memory measure for SecPri-BGMPOP threshold value “2”.

Table 3 lists the findings of the clustering-based time measurements for SecPri-BGMPOP threshold value 4 for WFCM. The graphs in Figures 11 and 12 show that when the cluster size increases, the time to complete increases since a high threshold requires more processing power and requires less memory.

Table 3. Clustering-Based Time and Memory Measures for SecPri-BGMPOP Threshold Value “4”.

<table>
<thead>
<tr>
<th>Number of Clusters</th>
<th>Time (s)</th>
<th>Memory (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22.37</td>
<td>1216</td>
</tr>
<tr>
<td>2</td>
<td>24.45</td>
<td>1216</td>
</tr>
<tr>
<td>4</td>
<td>25.64</td>
<td>1199</td>
</tr>
</tbody>
</table>
Based on clusters 1, 2, and 4, the Boost Graph Convolutional Network Clustering method was evaluated in terms of time and memory for the SecPri-BGMPOP threshold values of 1, 2, and 4. The suggested system and the Boost Graph Convolutional Network Clustering used in earlier work are then compared with FCM and WFCM in terms of time and memory requirements. Because our suggested approach is now maintained in a cloud environment, analytics consumers can use it without worrying about their security or privacy. In order to demonstrate that the method based on Boost Graph Convolutional Network Clustering has produced better results than the existing FCM methodology, we compared it to the latter. Measuring performance includes looking at things like memory utilisation and execution time. The Boost Graph Convolutional Network Clustering recommended method produced improved outcomes, as illustrated in Figure 13.

Figure 11. Clustering-based time measure for SecPri-BGMPOP threshold value “4”.

Figure 12. Clustering-based memory measure for SecPri-BGMPOP threshold value “4”.

Figure 13. Comparison of time measures with BGCNC and FCM for SecPri-BGMPOP threshold value “1, 2, 4”.
When comparing the Boost Graph Convolutional Network Clustering (BGCNC)-based SecPri-BGMPOP method to FCM clustering, it is discovered that Boost Graph Convolutional Network Clustering performs better. Here, we have examined time and memory across various cluster sizes. When we examine the data, we discover that the proposed system uses less memory and requires less time than the current FCM-based technique (see Figure 10). As a result, our suggested Boost Graph Convolutional Network Clustering approach performs better and produces better results.

The amount of time needed for computation in the key generation phase using the proposed MPBE, upgraded W09, Water 05, and MPBE curve. In Figures 14 and 15, compared to other methods, it can be seen that the Bibber approach requires less calculation time for the key creation, encryption, and decryption phases.

**Figure 14.** Comparison of proposed vs. FCM memory measures for SecPri-BGMPOP threshold value “1, 2, 4”.

**Figure 15.** Timing comparison graph for the proposed work in key generation, encryption phase, and decryption phase.

The simulation thus shows that our suggested information security technique outperformed the currently used traditional algorithms based on particular assaults. As a result, in Tables 4 and 5, it is evident from the simulation results that our SecPri-BGMPOP strategy outperforms other standard algorithms currently in use. Protecting this type of data is essential since it is necessary to use sensitive diagnostic information about autism to determine if a person is autistic. This type of information is more relevant in the healthcare industry. This study’s findings demonstrated that our suggested cleaning method protects these data against some threats more effectively than existing techniques. Nevertheless, it is indicated that, in terms of data security and privacy, the healthcare industry can make extensive use of our suggested strategy. Performance evaluation includes assessing memory utilisation and execution time. The SecPri-BGMPOP approach was shown to drastically reduce memory consumption and calculation time, which is consistent with the experiment’s findings.
Table 4. Sanitization Method and Proposed SecPri-BGMPOP.

<table>
<thead>
<tr>
<th>Performance Metrics</th>
<th>Proposed vs. Previous Approaches</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sanitization Method</td>
</tr>
<tr>
<td>Information loss</td>
<td>0.07%</td>
</tr>
<tr>
<td>Throughput</td>
<td>3.5 Mbps</td>
</tr>
<tr>
<td>Encryption time</td>
<td>3.625 Mbps</td>
</tr>
<tr>
<td>Decryption time</td>
<td>0.11 s</td>
</tr>
<tr>
<td>Efficiency</td>
<td>0.054 s</td>
</tr>
<tr>
<td></td>
<td>46.87 s</td>
</tr>
</tbody>
</table>

Table 5. Performance-Enhanced Hybrid Fragment Horde Bland Lobo Optimization.

<table>
<thead>
<tr>
<th>HFHBLO</th>
<th>PSO</th>
<th>GWO</th>
<th>Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Better than</td>
<td>0.26%</td>
<td>0.23%</td>
<td>CCA</td>
</tr>
<tr>
<td>Superior to</td>
<td>0.40%</td>
<td>0.29%</td>
<td>CPA</td>
</tr>
</tbody>
</table>

5. Conclusions

The SecPri-BGMPOP strategy aims, security, and privacy protection have been the subjects of distinct research endeavours in the past. The purpose of this essay was to provide big data analytics with security while maintaining privacy. The Boost Graph Convolutional Network Clustering (BGNC) algorithm, which reduces computational complexity in terms of time and memory measurements, is first applied to the input dataset to begin the clustering process. Second, it is enlarged by employing a piece of the magnifying bit string to generate a safe key; pinpointing-based encryption avoids amplify leakage even if a rival or attacker decrypts the key or asymmetric encryption. Finally, to determine the accuracy of the method, an optimal key was created using a meta-heuristic algorithmic framework called Hybrid Particle Swarm and Grey Wolf Optimisation (HPGWO). Our proposed method is currently kept in a cloud environment, allowing analytics users to utilise it without risking their privacy or security. We compared the technique based on Boost Graph Convolutional Network Clustering to the current FCM approach and created Boost Graph Convolutional Network Clustering to show that it has delivered superior results. Examining memory usage and execution time is part of measuring performance. Our SecPri-BGMPOP approach achieves an efficiency of 58.130 s with an information loss of 0.024%. The encryption and decryption times are 0.0086 s and 0.315 s, respectively. Confirming the experiment’s results, it was found that the SecPri-BGMPOP strategy significantly cuts down on memory usage and computation time. We intend to present additional real-world uses for the SecPri-BGMPOP scheme in the future and contrast it with other encryption techniques. In order to accelerate the encryption and decryption processes even more, we also aim to improve the SecPri-BGMPOP scheme.

Author Contributions: M.K. Conceptualization; J.P.A. initial draft of manuscript; H.-S.G. Experimental design and Methodology; B.P.K.: Validation and Analyses; J.S. Review; W.-C.L. Editing., Software. All authors have read and agreed to the published version of the manuscript.

Funding: Department of Electrical Engineering, Ming Chi University of Technology, Taiwan.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare that they have no potential conflicts.

References


36. Kavitha, C.; Srividhya, S.R.; Lai, W.-C.; Mani, V. IMapC: Inner MAPping Combiner to Enhance the Performance of MapReduce in Hadoop. *Electronics* 2022, 11, 1599. [CrossRef]


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.