Secure Rule Generation and Deep Computational Feature Learning model for Cloud Based Big Data

Ms. Prajakta Shirke[1], Prof. Shyam Gupta[2]
Department of Computer Engineering
Sidhant Collegen of Engineering Sudumbare, Pune 412109

Abstract- *In the work of big data analysis and its mining, feature learning plays a very important and fundamental role. The big data has various properties including volume, variety and velocity. Such properties become the challenging problems in the process of feature learning. Aiming to these problems, deep computational learning model is proposed here. This model is extended with the algorithm named as privacy preserving high order back propagation algorithm. The security of information is maintained during the deep computational over cloud big data, with the BGV fully homomorphism encryption scheme. In addition, rule generation method for frequent pattern generation is used with FP-Growth algorithm. The experimental results prove that the proposed system performs effective and efficient feature learning over cloud based big data.*

Keywords- Deep computational model, Privacy preservation, computational model, cloud, big data feature learning, encryption.

I. INTRODUCTION

Deploying the smart city is the key to improve the efficiency, reliability, and security of a traditional city. Smart city consists of intelligent transportation, smart grid, and intelligent security and so on. With the development of these fields, recent years have witnessed the remarkable growth of smart cities. With the massive deployment of various mobile devices, such as sensors and RFID, data are being collected at unprecedentedly rate in the smart city. To protect the private data and intermediate results, it requires secure computation of various operations needed by the deep computation model, including additions, multiplications, and the nonlinear Sigmoid function.

To improve the efficiency of deep computation model training and big data feature learning, it requires choosing the efficient full homomorphic encryption scheme according to the major operations of the algorithms in the privacy preserving deep computation model. To produce the correct result on the cipher-text using the full homomorphic encryption scheme, the sigmoid function is required to approximate as a new function involving only addition operations and multiplication operations. Therefore, it is critical for smart city planning, monitoring and controlling to develop big data modelling and analytic technologies.

As a fundamental technique of big data analytic, feature learning can discover the underlying structure of big data to provide intelligent decision for developing smart city systems. However, the characteristics of big data, referring to large scale of data, different types of data, and the speed of streaming data, pose feature learning many significant challenges. To tackle these challenges, we proposed a deep computation model for learning features on big data effectively in the previous work. Owning to the huge amount of data in the smart and high computational complexity, the deep computation model finds it difficult to perform in real time with limited computing power and memory storage.

Although the performance of computers has been improved, it still falls behind the growth of the big data size. Thus, how to support the real-time deep computation model training for big data feature learning is one of the most challenging issues in the smart city. The privacy conserving deep computation model poses variety of problems and challenges, particularly for large knowledge feature learning by incorporating the computing of the cloud. We tend to discuss the key challenges in 3 aspects as follows: to guard the non-public knowledge and intermediate results, it needs secure computation of varied operations required by the deep computation model, as well as additions, multiplications, and also the nonlinear Sigmoid perform.

To boost the potency of deep computation model coaching and massive knowledge feature learning, it needs to decide on the economical full homomorphic secret writing theme in step with the main operations of the algorithms within the privacy conserving deep computation model. to provide the proper result on the ciphertexts mistreatment the total homomorphic secret writing theme, the Sigmoid perform is needed to approximate as a brand new perform involving solely addition operations and multiplication operations. a privacy conserving deep computation model supported homomorphic secret writing.

The planned theme improves the potency by offloading the valuable computation tasks on the cloud. what is more, the planned theme prevents the revealing of the non-public mistreatment homomorphic secret writing that has been with success used for data processing and information discovery like call trees Bayesian networks support vector machines , and k-means .To support secure computation of varied operations like additions and multiplications needed by the high-order back-propagation algorithmic program, the paper encrypts the non-public knowledge mistreatment the BGV secret writing theme that's the presently most effective full homomorphic secret writing theme and supports at the same time supports discretional variety of addition operations and multiplication operations.

In this paper we study about the related work done, in section II, the proposed approach modules description, mathematical modeling, algorithm and experimental setup in section III .and at final we provide a conclusion in section IV.

II. LITERATURE REVIEW

In paper[1], Author proposed a privacy preserving deep learning model for big data feature learning by incorporating the computing power of the cloud. The proposed scheme uses the BGV encryption scheme to support the secure computation operations of the high-order back-propagation algorithm efficiently for deep computation model training on the cloud. In this scheme, only the encryption operations and the decryption operations are performed by the client while all the computation tasks are performed on the cloud. Experimental results demonstrated that the scheme can efficiently deal with deep computation model for big data feature learning by incorporating the high computing power of the cloud without disclosing private data. In addition with, the performance of scheme can be further improved by employing more cloud servers, which is particularly suitable for big data.

In paper[2], Author proposed the first secure and practical multi-party BPN network learning scheme over arbitrarily partitioned data. In this scheme, each party encrypts his/her private data locally and uploads the cipher texts into the cloud. Without knowing the original private data the cloud executes most of the operations pertaining to the learning algorithms over cipher texts. The cost of each party in this scheme is independent to the number of parties. This work tailors the BGN homomorphic encryption algorithm to support the multi-party scenario, which can be used as an independent solution for other related applications.

In paper[3],Author stated as the challenges of preserving privacy in cloud storage. To address these challenges, apply oblivious RAM (ORAM), known to be the most effective solution for hiding user access patterns. It provide a tutorial about ORAM and survey recent efforts to increase the practicability of using it in cloud storage by reducing its overhead. Consider distributed file systems built on hundreds or thousands of servers in a single or multiple geo distributed cloud sites. Applying an ORAM-based algorithm for privacy-preserving access can lead to serious access load imbalance among the storage servers. Author propose a low-complexity algorithm that can deal with a large-scale problem with respect to big data. It conduct extensive simulations to show that the proposed algorithm finds results close to the optimal solution, and significantly outperforms a random data-placement algorithm.

In paper [4] authors claims to propose that they have developed the first secure and practical multi-party BPN network learning scheme over arbitrarily partitioned data. In developed approach, the parties encrypt their arbitrarily partitioned data and upload the cipher texts to the cloud. The cloud can execute most operations pertaining to the BPN network learning algorithm without knowing any private information. The cost of each party in our scheme is independent to the number of parties. This work tailors the BGN homomorphism encryption algorithm to support the multi-party scenario, which can be used as an independent solution for other related applications.

In paper[5], Author propose an efficient and fine-grained big data access control scheme with privacy-preserving policy. Specifically, it hide the whole attribute (rather than only its values) in the access policies. To assist data decryption, Author design a novel Attribute Bloom Filter to evaluate whether an attribute is in the access policy and locate the exact position in the access policy if it is in the access policy. Security analysis and performance evaluation show that the scheme can preserve the privacy from any LSSS access policy without employing much overhead.

In paper[6], Author propose an efficient and flexible protocol, called EFPA, for privacy-preserving association rule mining in cloud. With the protocol, plenty of participants can provide their data and mine the association rules in cloud together without privacy leakage. Detailed security analysis shows that the proposed EFPA protocol can achieve privacy-preserving mining of association rules in cloud. In addition, performance evaluations via extensive simulations also demonstrate the EFPA’s effectiveness in term of low computational costs.

In paper [2] authors developed a privacy preserving back propagation algorithm depending on the BGV encryption technique on cloud. One property of the designed algorithm is to apply the BGV encryption system to the back-propagation algorithm for preventing the disclose of private data with cloud computing. Furthermore, the developed algorithm improved the efficiency of massive data feature learning by incorporating the strong power of the cloud computing.

In paper[8], Author proposed a format-compliant end-to-end privacy-preserving scheme for media sharing/storage issues with considerations for big data, clouds, and mobility. To realize efficient encryption for big media data, jointly achieve format-compliant, compression-independent and correlation-preserving via multichannel chained solutions under the guideline of Markov cipher. The encryption and decryption process is integrated into an image/video filter via GPU Shader for display-to-display full encryption. The proposed scheme makes big media data sharing/storage safer and easier in the clouds.

III. PROPOSED APPROACH

A. Problemnn Statement

The privacy preserving deep computation model poses a number of issues and challenges, especially for big data feature learning by incorporating the computing of the cloud.

- To protect the private data and intermediate results, it requires secure computation of various operations needed by the deep computation model, including additions, multiplications, and the nonlinear Sigmoid function.
- To improve the efficiency of deep computation model training and big data feature learning, it requires to choose the efficient full homomorphic encryption scheme according to the major operations of the algorithms in the privacy preserving deep computation model.
- To produce the correct result on the cipher texts using the full homomorphic encryption scheme, the Sigmoid function is required to approximate as a new function involving only addition operations and multiplication operations

B. Proposed System Overview

In propose system user give Performance Measurement System that is PeMS dataset as a input to the server. Then server extract the features from dataset by applying Tensor Auto Encoder (TAE) method. Then apply BGV encryption algorithm to protect the private data. Also we use C4.5 classification algorithm to classify the encrypted features. Then predict the data and apply rule generation approach that is FP-Growth algorithm to generate the frequent patterns and give it to user.

1. Input Dataset
A real dataset named as Performance Measurement System (PeMS) is used to evaluate the performance of system. This dataset is used for prediction of traffic flow.

2. Feature Learning with TAE
Tensor auto-encoders are used at user side, to learn the features from Big data.

3. BGV Encryption
The extracted features are then encrypted with BGV scheme of homomorphic encryption. This is used to secure the data of users. Security is required because, this data is outsourced to cloud server, for further operations.

4. Arithmetic Operations at Cloud Server
After receiving encrypted features, cloud servers apply privacy preserving high-order back-propagation algorithm on encrypted features. That is arithmetic operations are performed for parameter updating. This enhanced the security.

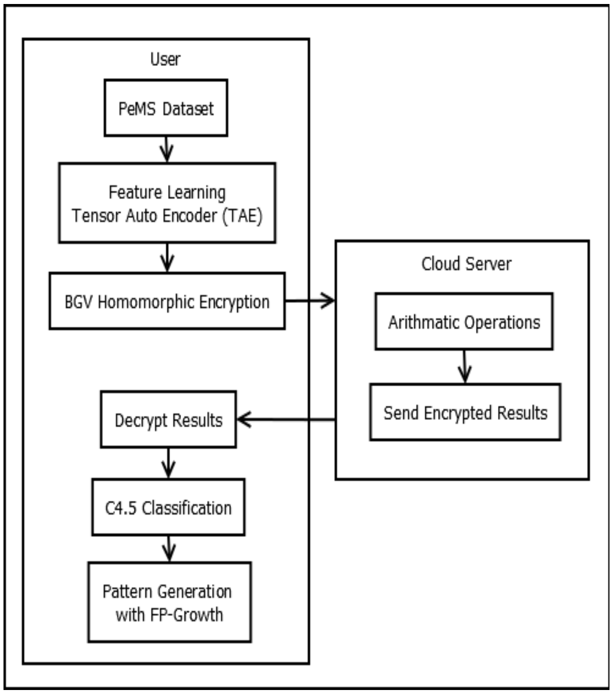


Figure 1.Proposed System Architecture

6. C4.5 Classification
After receiving the output from cloud server, user decrypt that results and perform classification with C4.5 algorithm.

7. Pattern Generation with FP-Growth
From classified results, rules / patterns are generated by using FP-Growth algorithm. A rule generation approach is used to generate the frequent patterns from the predicted result by using FP-Growth algorithm.

C. Algorithm

Algorithm 1: FP-Growth Algorithm

Input: A database DB and a minimum support threshold.
Output: The complete set of frequent patterns.

Process: call FP-growth(FP-tree, null).

```
Procedure FP-growth(Tree, a) {
    if Tree contains a single prefix path then { // Mining single prefix-path FP-tree
        let P be the single prefix-path part of Tree;
        let Q be the multipath part with the top branching node replaced by a null root;
        for each combination (denoted as A˜ ) of the nodes in the path P do generate pattern $\beta \cup$ a with support = minimum support of nodes in $\beta$;
        let freq pattern set(P) be the set of patterns so generated;
    }
    else let Q be Tree;
    for each item ai in Q do { // Mining multipath FP-tree generate pattern $\beta$ = ai $\cup$ a with support = ai .support;
        construct $\beta$ ’s conditional pattern-base and then ˜Aaˆs conditional FP-tree Tree A˜ ; if Tree then $\beta \neq \theta$
        call FP-growth(Tree $\beta$ , $\beta$);
        let freq pattern set(Q) be the set of patterns so generated;
    }
    return(freq pattern set(P) $\cup$freq pattern set(Q) $\cup$ (freq pattern set(P) x freq pattern set(Q)))
}
```

D. Mathematical Model

Input: PeMS Dataset
Output: Prediction of patterns

Process:

1. Feature Learning

For extraction of feature from big data, TAE is used.
F = {f1, f2, …, fn}
Where, F is the set of n number of extracted features from dataset.

2. Feature Encryption

Extracted features are encrypted using BGV homomorphic encryption scheme.
EF = {ef1, ef2, …, efn}
Where, EF is the set of encrypted features.

3. Arithmetic operations

After outsourcing of data on cloud server, arithmetic operations are performed.
A = {a, s, m, d, e}
Where,
a = Addition
s = Subtraction
m = Multiplication
d = Division
e = Exponentiation

4. Classification

For Classification, C4.5 classifier is used.
C = {c1, c2, …., cn}
Where, C is the set of classified results with particular label.

5. Pattern Prediction

For prediction of patterns, FP-Growth algorithm is used.
P = {p1, p2, …., pn}
Where, P is the set of predicted patterns,

IV. RESULTS AND DISCUSSION

A. Experimental Setup

The system is built using Java framework on Windows platform. The Net beans IDE is used as a development tool. The system doesn’t require any specific hardware to run; any standard machine is capable of running the application.

B. Dataset

A real dataset named as Performance Measurement System (PeMS) is used to evaluate the performance of system. This dataset is used for prediction of traffic flow [11].

C. Evaluation Result

Table 1depicts the time required, memory and accuracy of C4.5 and Rule generation classifiers. Time is measured in terms of milliseconds, memory is measured in terms of bytes and accuracy is measured in terms of bytes.

Performance Measures	C4.5 Classifier	Rule Generation Classifier
Time in ms	280	160
Memory in bytes	245234598	107354297
Accuracy in %	70	89

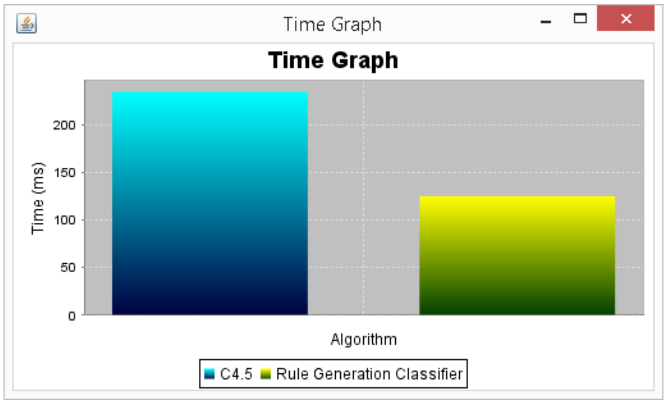


Fig. 2: Graph of Classification Time

Figure 2 represent the graphical comparison of C4.5 and rule generation classifier. From this graph it is clear that rule generation classifier is more time efficient than the C4.5 classifier.

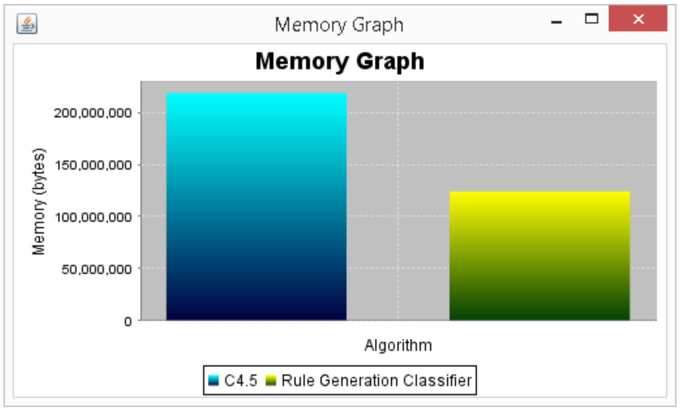


Fig . 3: Graph of Classification Memory

Figure 3 represent the graphical comparison of C4.5 and rule generation classifier based on memory. From this graph it is clear that rule generation classifier consumes less memory than the C4.5 classifier.

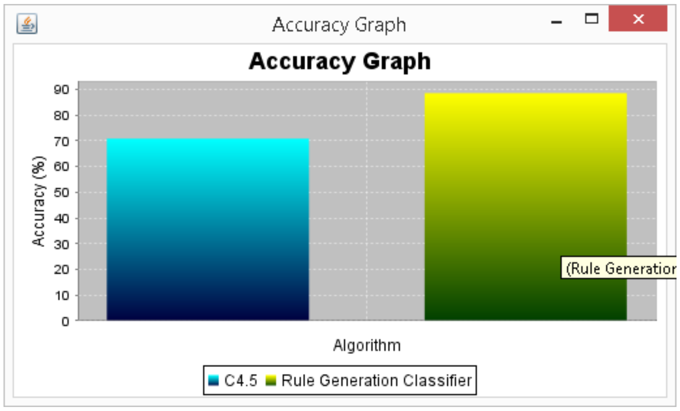


Fig. 4: Graph of Classification Accuracy

Figure 4 represent the graphical comparison of C4.5 and rule generation classifier on the basis of accuracy parameters. From this graph it is clear that rule generation classifier is more accurate than the C4.5 classifier.

D. System Screenshots

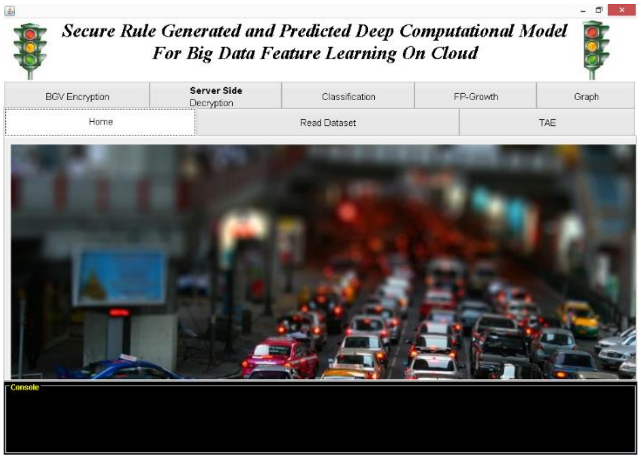
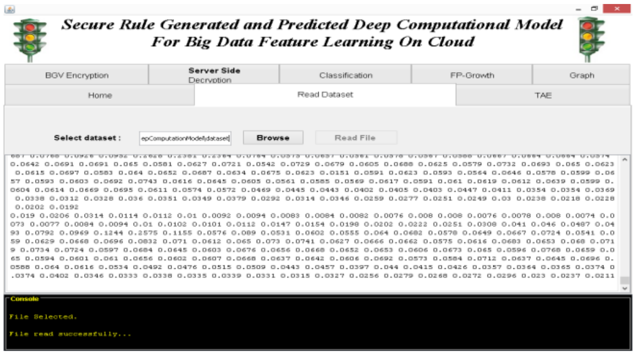

Secure Rule Generated and Predicted Deep Computational Model
For Big Data Feature Learning On Cloud
BGV Encryption
Server Side
Decryption
Classification
FP-Growth
Graph
Home
Read Dataset
TAE
Console

Fig 5.Home page.



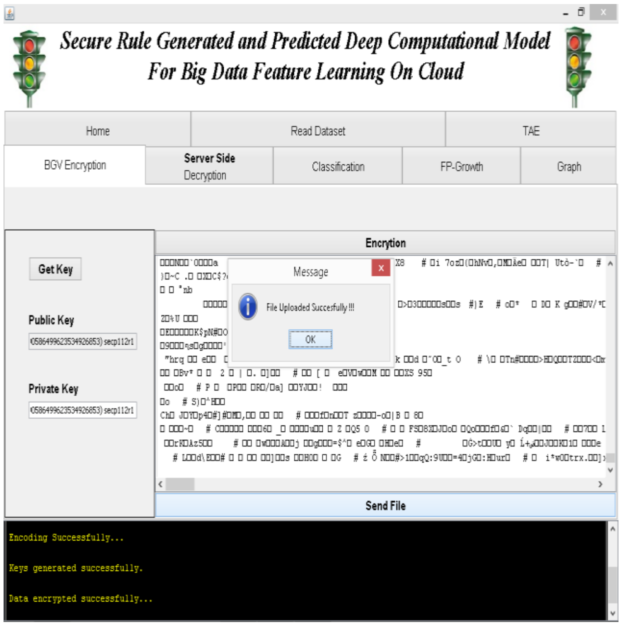
Select dataset : epComputationModel\dataset

```
0.0642 0.0691 0.0691 0.065 0.0581 0.0627 0.0721 0.0542 0.0729 0.0679 0.0605 0.0688 0.0625 0.0579 0.0732 0.0693 0.065 0.0623
 0.0615 0.0697 0.0583 0.064 0.0652 0.0687 0.0634 0.0675 0.0623 0.0151 0.0591 0.0623 0.0593 0.0564 0.0646 0.0578 0.0599 0.06
57 0.0593 0.0603 0.0692 0.0743 0.0616 0.0645 0.0605 0.0561 0.0585 0.0569 0.0617 0.0591 0.061 0.0619 0.0612 0.0639 0.0599 0.
0604 0.0614 0.0669 0.0695 0.0611 0.0574 0.0572 0.0469 0.0445 0.0443 0.0402 0.0405 0.0403 0.0447 0.0411 0.0354 0.0354 0.0369
 0.0338 0.0312 0.0328 0.036 0.0351 0.0349 0.0379 0.0292 0.0314 0.0346 0.0259 0.0277 0.0251 0.0249 0.03 0.0238 0.0218 0.0228
 0.0202 0.0192
0.019 0.0206 0.0314 0.0114 0.0112 0.01 0.0092 0.0094 0.0083 0.0084 0.0082 0.0076 0.008 0.008 0.0076 0.0078 0.008 0.0074 0.0
073 0.0077 0.0084 0.0094 0.01 0.0102 0.0101 0.0112 0.0147 0.0154 0.0198 0.0202 0.0222 0.0251 0.0308 0.041 0.046 0.0487 0.04
93 0.0792 0.0969 0.1244 0.2575 0.1155 0.0576 0.089 0.0531 0.0602 0.0555 0.064 0.0682 0.0578 0.0649 0.0667 0.0724 0.0541 0.0
59 0.0629 0.0668 0.0696 0.0832 0.071 0.0612 0.065 0.073 0.0741 0.0627 0.0666 0.0662 0.0575 0.0616 0.0683 0.0653 0.068 0.071
9 0.0734 0.0724 0.0597 0.0684 0.0645 0.0603 0.0676 0.0656 0.0668 0.0652 0.0653 0.0606 0.0673 0.065 0.0596 0.0768 0.0659 0.0
65 0.0594 0.0601 0.061 0.0656 0.0602 0.0607 0.0668 0.0637 0.0642 0.0606 0.0692 0.0573 0.0584 0.0712 0.0637 0.0645 0.0696 0.
0588 0.064 0.0616 0.0534 0.0492 0.0476 0.0515 0.0509 0.0443 0.0457 0.0397 0.044 0.0415 0.0426 0.0357 0.0364 0.0365 0.0374 0
.0374 0.0402 0.0346 0.0333 0.0338 0.0335 0.0339 0.0331 0.0315 0.0327 0.0256 0.0279 0.0268 0.0272 0.0296 0.023 0.0237 0.0211
```

```
File Selected.

File read successfully...
```

Fig.6 Read Dataset



Public Key: 0586499623534926853) secp112r1

Private Key: 0586499623534926853) secp112r1

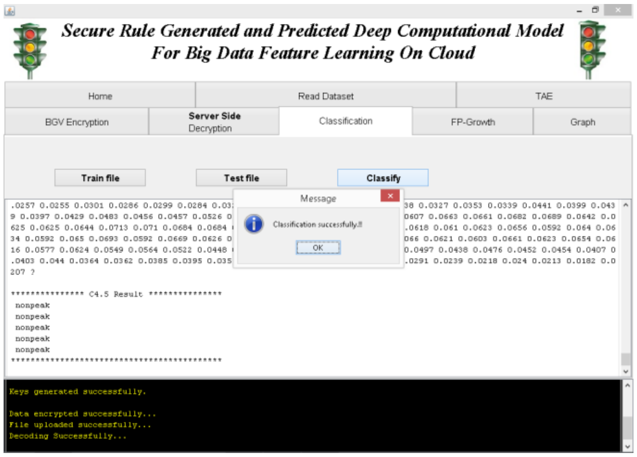
File Uploaded Succesfully !!!

```
Encoding Successfully...

Keys generated successfully.

Data encrypted successfully...
```

Fig.7 File Uploading



Classification successfully.!!

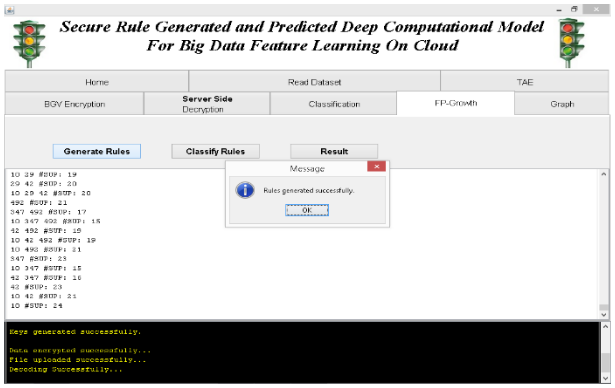
```
.0257 0.0255 0.0301 0.0286 0.0299 0.0284 0.03                       38 0.0327 0.0353 0.0339 0.0441 0.0399 0.043
9 0.0397 0.0429 0.0483 0.0456 0.0457 0.0526 0                        0607 0.0663 0.0661 0.0682 0.0689 0.0642 0.0
625 0.0625 0.0644 0.0713 0.071 0.0684 0.0684                         .0618 0.061 0.0623 0.0656 0.0592 0.064 0.06
34 0.0592 0.065 0.0693 0.0592 0.0669 0.0626 0                        066 0.0621 0.0603 0.0661 0.0623 0.0654 0.06
16 0.0577 0.0624 0.0549 0.0564 0.0522 0.0448                         0.0497 0.0438 0.0476 0.0452 0.0454 0.0407 0
.0403 0.044 0.0364 0.0362 0.0385 0.0395 0.035                        .0291 0.0239 0.0218 0.024 0.0213 0.0182 0.0
207 ?

*************** C4.5 Result ***************
 nonpeak
 nonpeak
 nonpeak
 nonpeak
 nonpeak
*******************************************
```

```
Keys generated successfully.

Data encrypted successfully...
File uploaded successfully...
Decoding Successfully...
```

Fig 8. Classification



Rules generated successfully.

```
10 29 #SUP: 19
29 42 #SUP: 20
10 29 42 #SUP: 20
492 #SUP: 21
347 492 #SUP: 17
10 347 492 #SUP: 15
42 492 #SUP: 19
10 42 492 #SUP: 19
10 492 #SUP: 21
347 #SUP: 23
10 347 #SUP: 15
42 347 #SUP: 16
42 #SUP: 23
10 42 #SUP: 21
10 #SUP: 24
```

```
Keys generated successfully.

Data encrypted successfully...
File uploaded successfully...
Decoding Successfully...
```

Fig 9. Rule Generation

V.CONCLUSION AND FUTURE SCOPE

As more applications are attracting towards cloud, it is important to study the big data over cloud for feature learning and pattern generation. This paper develops the deep computational model over cloud along with maintaining the privacy. Further system is combined with the feature learning process. For feature learning TAE approach is used. These entire features are store on cloud with maintaining privacy and security with the help of BGV encryption scheme. These features are then go through computational model. System makes use of C4.5 classifier fro classification of test data and FP-growth algorithm for pattern discovery. The performance of system is tested with PeMS dataset and experimental results prove that the system achieves higher prediction accuracy than existing one. For future work we recommended to use this system for smart city applications.

REFERENCES

[1] Zhang, Qingchen, Laurence T. Yang, and Zhikui Chen. "Privacy Preserving Deep Computation Model on Cloud for Big Data Feature Learning." IEEE Transactions on Computers 65.5 (2016): 1351-1362.

[2] Yuan, Jiawei, and Shucheng Yu. "Privacy preserving back-propagation neural network learning made practical with cloud computing." IEEE Transactions on Parallel and Distributed Systems 25.1 (2014): 212-221.

[3] Li, Peng, et al. "Privacy-Preserving Access to Big Data in the Cloud." IEEE Cloud Computing 3.5 (2016): 34-42.

[4] J. Yuan and S. Yu, “Privacy Preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing,” in IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 1, pp. 212-221, Jan. 2014.