e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

CORRELATION REDUCTION SCHEME FEATURE SELECTION USING MACHINE LEARNING

Dr.Saravanan.K*[1], Sumanth.K*[2], Chaitanya Yadav.S*[3], Vamshidhar Reddy.P*[4], Thiran Kumar Reddy.A*[5]

*[1]Assistant Professor, Department Of Computer Science And Engineering, Madanapalle Institute Of Technology And Science, Madanapalle, India.

*[2,3,4,5]B. Tech IV Year, Department Of Computer Science And Engineering, Madanapalle Institute Of Technology And Science, Madanapalle, India.

ABSTRACT

In industrial areas, preserving privacy is crucial since data used for training in industries contains important data. A lack of consideration of data correlation in machine learning algorithms on differentially private data may lead to better privacy leaks than expected in industrial applications. For instance the data collected for traffic monitoring contains some correlated records because of temporal correlation or user correlation. The solution to this problem is to propose a correlation reduction scheme that considers the privacy loss of data correlated in machine learning tasks. This scheme uses five steps to tackle the problem of managing the degree of data correlation, preserving privacy, and ensuring accuracy of the predictions. The proposed feature selection scheme thus eliminates the impact of data correlation, while ensuring that the privacy issue associated with data correlation is avoided. The method in this proposal can be applied to almost any machine learning algorithm that provides services in industry. According to experiments it has shown that the proposed scheme can produce better prediction results with machine learning tasks and less mean square errors for data queries than previous schemes.

Keywords: Differential Privacy, Machine Learning, Data Correlation, Feature Selection.

I. INTRODUCTION

Nowadays, The Internet of Things (IoT) and smart cities are among the industrial applications where machine learning has become an indispensable tool. Machine learning in industry relies heavily on activities of humans as a data source. For example , Smart phones represent one tool for collecting and analyzing human data to provide urban services such as traffic monitoring and smart health information. The above example and anecdote about location and health are two examples of information collected from humans that can contain sensitive information. Individual privacy can be compromised if these data are used for machine learning. The differential privacy technique was initially proposed by Dwork et al., and it is a popular method for protecting privacy. Because differential privacy provides a mathematical framework for protecting privacy, differential privacy has gained considerable attention since then. Differential privacy technology is being used to protect the privacy of people in a variety of industrial information technologies like smart grids, intelligent telematics, and multi-agent systems. There has been a great deal of work done on differential privacy in machine learning. Using a specific linear perturbation item, Chaudhuri provided differentially private stochastic gradient descent mechanisms and an empirically tested output perturbation model based on trained and noise-labelled output perturbations. On the basis of a differentially private variation of stochastic gradient descent, we have proposed a deep learning algorithm. Differences in private machine learning algorithms have not been used in previous work as data correlated with algorithms.

The proposed CR-FS scheme:

Using the algorithm I with differential privacy as the basis for selection, we proceed to select features traditionaly. When analyzing a dataset, selecting features is an important part of the machine learning process, especially when dealing with data rich in attributes. As a result, retaining more features can result in higher levels of data correlation, which can negatively affect privacy levels. Thus, we select features in a way that minimizes the correlation of data between them, while still maintaining excellent functionality for data publishing and analysis.

Literature Survey:

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

There has been little success in the use of location privacy protection techniques in the big data environment, including the sensor networks that contain sensitive data that must be adequately secured, based on the traditional techniques of anonymization, fuzzy computing, and cryptography. As a result of new trends such as Industry 4.0 and the Internet of Things (IoT), huge amounts of sensitive data is generated, processed, and exchanged, making them attractive targets for cyber attacks. Earlier methods, however, failed to take into account the need to protect privacy, leading to privacy violations. Using index mechanisms, we propose a method that protects location data privacy in Industrial Internet of Things and maximizes the utility of data and algorithms by meeting differential privacy constraints. Similarly, differential privacy is applied to select data by considering the frequency of node accesses to the tree. Using low density location data and high utility value they provide, this model combines the privacy and utility of location information. Moreover, the differential privacy index mechanism selects data based on the frequency of accessing tree nodes. The last step is to add noise to the frequency with which the accessing data are selected by applying the Laplace scheme. This research concludes that the proposed strategy improves security, privacy, and applicability in addition to delivering significant benefits.

Algorithm:

Pseudo Code for Feature Selection Algorithm (Sigmis) Input : S(F1, F2, ……., Fk, Fc) // a training data set

Output : Sbest // the selected feature set

Step 1 : begin

Step 2 : for i = 1 to k do begin

r = calculate correcoeff(Fi , Fc);

end;

// let p = 0.05 significant level;

Step 3 : let ρ = 0 // assuming there is no significant correlation between Fi and Fc ;

Step 4 : for i = 1 to k do begin

t = calculate signi(r, ρ) for Fi ; // using t-test value from eqn (2)

if t > CV // Critical value

Sbest = Slist;

end;

Step 5 : return Sbest

; end;

Pseudo code for handling missing values

Let F be the set of features with {F1, F2, F3 , ……., Fk, Fc } where k is the number of features and Fc is the class feature. Let Slist be the training data set with n instances and Smiss be the set of missing values {m1, m2, …..ms }, where s is the number of missing values. Let V = {v1, v2…., vn} be the instances. Take a missing value from Smiss and find its value with Slist. Assume the fixed value for the missing value (say 10) to find the correct missing value. Do the same process for all the missing values. To do this, first sort the data set along with the class feature.

Input : Slist(F1, F2, ……., Fk , Fc) // a training data set

Output : Snew // the changed data set

Step 1 : begin

Step 2 : for i = 1 to k do begin

Sort the feature Fi in S along with the class feature Fc ;

end;

Step 3 : for i = 1 to n do begin

If (vi = = NULL) // if there is a missing value

Smiss = Slist ;

break;

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

end;

Step 4 : for i = 1 to length(Smiss) do begin append(Slist)=getFirstElement(Smiss)
Smiss = vi − vi−1 // Find the missing value by finite difference method;

end;

replace the missing value by using the formula miss_value = abs(Smiss − fix)

Step 5 : repeat Step 3 and 4 for the remaining features.

Step 6 : return Snew

Step 7 : end;

Screenshots:

```
from sklearn.linear_model import LogisticRegression
import warnings
warnings.filterwarnings('ignore')
lr = Pipeline([
    ('scaler', MinMaxScaler()),
    ('clf', dp.LogisticRegression(epsilon=5))
])

lr.fit(X_train, y_train)
```

```
Pipeline(memory=None,
         steps=[('scaler', MinMaxScaler(copy=True, feature_range=(0, 1))),
                ('clf',
                 LogisticRegression(C=1.0, data_norm=3.0003767724634693,
                                    epsilon=5, fit_intercept=True, max_iter=100,
                                    n_jobs=None, tol=0.0001, verbose=0,
                                    warm_start=False))],
         verbose=False)
```

```
from sklearn.metrics import accuracy_score
y_pred_lr = lr.predict(X_test)
print('Acc :',accuracy_score(y_test,y_pred_lr))
```

```
Acc : 0.8596491228070176
```

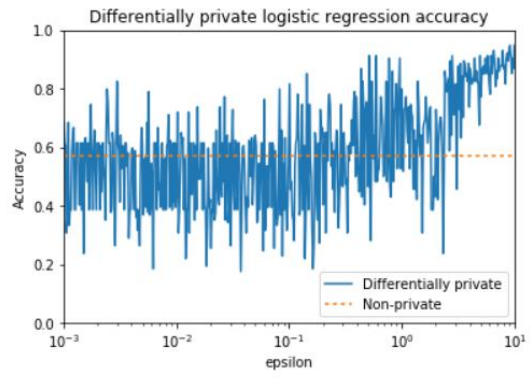
```
dp_lr = Pipeline([
    ('scaler', MinMaxScaler()),
    ('clf', dp.LogisticRegression(epsilon=4))
])

dp_lr.fit(X_train, y_train)
```

```
Pipeline(memory=None,
         steps=[('scaler', MinMaxScaler(copy=True, feature_range=(0, 1))),
                ('clf',
                 LogisticRegression(C=1.0, data_norm=2.821762643727689,
                                    epsilon=4, fit_intercept=True, max_iter=100,
                                    n_jobs=None, tol=0.0001, verbose=0,
                                    warm_start=False))],
         verbose=False)
```

```
from sklearn.metrics import accuracy_score
y_pred_dp_lr = dp_lr.predict(X_test)
print('Acc :',accuracy_score(y_test,y_pred_dp_lr))
```

```
Acc : 0.8596491228070176
```



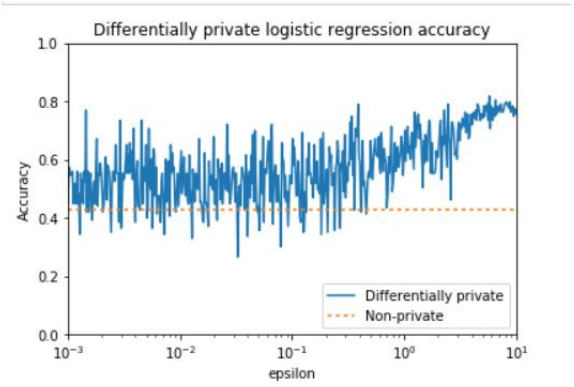

```
from sklearn.linear_model import LogisticRegression
import warnings
warnings.filterwarnings('ignore')
lr = Pipeline([
    ('scaler', MinMaxScaler()),
    ('clf', dp.LogisticRegression(epsilon=10))
])

lr.fit(X_train, y_train)
```

```
Pipeline(memory=None,
         steps=[('scaler', MinMaxScaler(copy=True, feature_range=(0, 1))),
                ('clf',
                 LogisticRegression(C=1.0, data_norm=3.3369691384787403,
                                    epsilon=10, fit_intercept=True,
                                    max_iter=100, n_jobs=None, tol=0.0001,
                                    verbose=0, warm_start=False))],
         verbose=False)
```

```
from sklearn.metrics import accuracy_score
y_pred_lr = lr.predict(X_test)
print('Acc :',accuracy_score(y_test,y_pred_lr))
```

```
Acc : 0.7692307692307693
```



II. CONCLUSION

As discussed in the present paper, machine learning has a privacy problem due to data correlation that may result in more damaging privacy losses in industrial applications than expected. An innovative feature selection scheme is proposed to reduce data correlation with little impact on data utility. There are steps in the proposed system that consider the accuracy of predicted results, preserving privacy, and analyzing the correlation of the data in each dataset. Compared to current approaches, our new algorithm is better at balancing privacy leaks and data utility. Several experiments have been conducted to test the method's performance, and we confirm that our CR-FS scheme can provide either better data analysis or better data queries than other schemes.

III. REFERENCE

[1] U.S.Shanthamallu,A.Spanias,C.TepedelenliogluandM.Stanley,"Abrief survey of machine learning methods and their sensor and IoT applications," In 2017 8th International Conference on Information, Intelligence, Systems and Applications (IISA), pp. 1-8.

[2] I.A.T. Hashem, V. Chang, N.B. Anuar, K. Adewole, I. aqoob, A. Gani, E. Ahmed and H. Chiroma, "The role of big data in smart city," International Journal of Information Management, 36(5), pp.748-758.

[3] C. Yin, J. Xi, R. Sun and J. Wang, "Location privacy protection based on differential privacy strategy for big data in industrial internet of things," IEEE Transactions on Industrial Informatics, 2017, 14(8), pp.3628-3636.

[4] A. Solanas, C. Patsakis, M. Conti, I. Vlachos, V. Ramos, F. Falcone, O. Postolache, P. Perez-Martinez, R. Pietro, D. Perrea, "Smart health: a context-aware health paradigm within smart cities," IEEE Communications Magazine, vol. 52, no. 8, pp. 74–81.

[5] C.M. Benjamin, M. Fung, K. Wang, R. Chen and P.S. Yu, "Privacy- preserving data publishing: A survey of recent developments," ACM Computing Surveys, 2010, 42(4), pp.1-53. [6] C. Dwork, 2006, "Differential privacy," in ICALP, pp. 1–12.

[6] M. Yang, T. Zhu, Y. Xiang and W. Zhou, 2018. "Density-based location preservation for mobile crowd sensing with differential privacy," IEEE Access, 2018, 6, pp.14779-14789.

[7] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: privacy preserving fog-enabled aggregation in smart grid," IEEE Transactions on Industrial Informatics, 2018, 14(8), pp.3733-3744.

[8] Y. Liu, W. Guo, C.I. Fan, L. Chang and C. Cheng, "A practical privacy- preserving data aggregation (3PDA) scheme for smart grid," IEEE Transactions on Industrial Informatics, 2019, 15(3), pp.1767-1774.

[9] D. Ye, T. Zhu, W. Zhou, and P.S. Yu, "Differentially Private Malicious Agent Avoidance in Multiagent Advising Learning," IEEE transactions on cybernetics, 2019, DOI:10.1109/TCYB.2019.2906574.