JOURNAL OF BASIC SCIENCE AND ENGINEERING

FRAUDULENT ACTIVITY DETECTION IN ONLINE SHOPPING USING MACHINE LEARNING

[1]Shaik Shakila Banu, [2]K. Chandra Prasad

[1]PG Scholar, Department of CSE, Srinivasa Institute of technology and science, Kadapa.
[2]Assistant Professor, Department of CSE, Srinivasa Institute of technology and science, Kadapa.
[1]shakila912banu@gmail.com, [2]chandu.sp2009@gmail.com

ABSTRACT

The volume of internet users is increasingly causing transactions on e-commerce to increase as well. We observe the quantity of fraud on online transactions is increasing too. Fraud prevention in e-commerce shall be developed using machine learning, this work to analyse the suitable machine learning algorithm, the algorithm to be used is the Decision Tree, Naive Bayes, Random Forest, and ensembled algorithms. Large amounts of money are often handled on e-commerce websites. And when large amounts of money are moved, there is a high risk that users will engage in fraudulent activities, e.g. Eg Use of stolen credit cards, money laundering, etc.

Keywords: Fraud Detection, Machine Learning, E-Commerce, Decision Tree, Naive Bayes, Random Forest, Fraud Prevention.

I. INTRODUCTION

The growth of e-commerce sector, the count of e-commerce related frauds are also increasing in every year since 1993. As per a report in 2013, 5.65 cents are lost due to frauds out of every $100 in ecommerce turnover. Fraud detection [1] is a key area requiring attention to avoid business losses and to uphold the consumer trust [2] [3]. Most observed frauds in e-commerce industry include stolen credit or debit card information and fraudulent return of products. Over the period of time, researchers have come up with different strategies [4] to detect card related fraudulent actions. The key strategies evolved include Artificial Immune Systems [5], Use of Periodic Features [6], Inductive Learning and Evolutionary Algorithm [7], Hidden Markov Model [8], Neural Data Mining [9], Fusion Approach [10], Bayes Mini-mum Risk Algorithm [11], etc. Another type of fraud which has become prominent with the evolution of marketplaces is the merchant fraud. These frauds are directly impacting customer satisfaction level and thereby reducing the trustworthiness of the marketplace itself [12] [13]. So, marketplace owners are keen in terms of identifying such fraudulent sellers. With the evolution of big data, data mining and machine learning techniques, it is possible to perform analysis on the historic data and correlate it with seller behaviours to identify potential fraudulent moves. The proposed model results in proactive identification of fraudulent selling attempts in a marketplace with the help of machine learning strategies.

II. BACKGROUND WORK

Fraud detection that has developed very rapidly is fraud on credit cards. Many studies discuss the fraud method. One of the studies carried out using deep learning is auto-encoder and restricted

Boltzmann machine [9]. Deep learning is used to build a fraud detection model that runs like a human neural network, where data will be made in several layers that are tiered for the process, starting from the Encoder at layer 1 hinge decoder at layer 4. The researcher compares the deep learning method with other algorithms such as Hidden Markov Model (HMM) [10]. Credit card fraud detection research was also using machine learning [11] machine learning used as a decision tree algorithm, naïve Bayes, neural networks, and random forests.

Decision tree is one algorithm that is widely used in fraud detection because it is easy to use. Decision tree is a prediction model using tree structure or hierarchical structure. Naïve Bayes is used in fraud detection credit cards because Naïve Bayes is a classification with probability and statistical methods. Naïve Bayes is very fast and quite high inaccuracy in real-world conditions neural network on fraud detection credit cards uses genetic algorithms to determine the number of hidden layer architectures on neural networks [12] with genetic algorithm, the genetic algorithm produces the most optimal number of hidden layers [13]. Fraud detection on credit cards also uses random forest [14]. Random forest uses a combination of each good tree and then combined into one model. Random Forest relies on a random vector value with the same distribution on all trees where each decision tree has a maximum depth [15]. Research on fraud detection in e-commerce is still not much so far. Fraud detection research on e-commerce is only limited to the determination of features or attributes that will be used to determine the nature of the fraud or non-fraud transactions [16]. The study describes the extraction. attribute/feature process used to determine behavior in ecommerce transactions. This attribute is used as fraud detection in e-commerce.

This attribute determines the transaction conditions. Another research on fraud detection in e-commerce is a reason transaction based on the attributes or features that exist in e-commerce transactions. The features/attributes used are features of the transaction, namely invalid rating, confirmation interval, average stay time on commodities, a feature of buyer namely real name, positive rating ratio, transaction frequency. Imbalance of data results in suboptimal classification results. The dataset on the paper has a total number of 151,112 records, the dataset classified as fraud is 14,151 records, and the ratio of fraud data is 0.093 percent. Synthetic Minority Oversampling Technique (SMOTE) is one of the methods used to make data into balance, Synthetic Minority Oversampling Technique (SMOTE) [17] is one of the oversampling methods that work by increasing the number of positive classes through random replication of data, so that the amount of data positive is the same as negative data. The way to use synthetic data is to replicate data in a small class. The SMOTE algorithm works by finding k closest Neighbor for a positive class, then constructing duplicate synthetic data as much as the desired percentage between randomly and positively chosen k classes.

III. SYSTEM ANALYSIS

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of software development methodologies [4].

i) Existing System

- Fraud detection that has developed very rapidly is fraud detection on credit cards ranging from fraud detection using machine learning to fraud detection using deep learning but unfortunately fraud detection for transactions on e-commerce is still small, fraud detection research on e-commerce commerce is still not much so far, fraud detection research on e-commerce is only limited to the determination of features or attributes.
- Research on fraud detection in e-commerce is still not much so far. Fraud detection research on e-commerce is only limited to the determination of features or attributes that will be used to determine the nature of the fraud or non-fraud transactions.

ii) Proposed System

- In this project by using different technique that is being used to execute online transaction fraud detection impact on the online shopping website as well as merchant and customer.
- This machine-learning algorithm will be compared to find the best accuracy results from the transaction dataset in e-commerce.
- Machine learning is really great at detecting fraudulent activity. Every website that you provide your credit card information to has a risk team that is responsible for preventing machine learning fraud. The goal of this challenge is to create a machine learning model that predicts the probability that a new user's first transaction will be fraudulent.
- Company XYZ is an e-commerce site that sells handmade clothing. We need to build a model that predicts whether a user is likely to use the site for illegal activities. We only have information about the first transaction of the user on the website and based on that, you must perform your classification ("fraud / no fraud").

IV. SYSTEM DESIGN

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

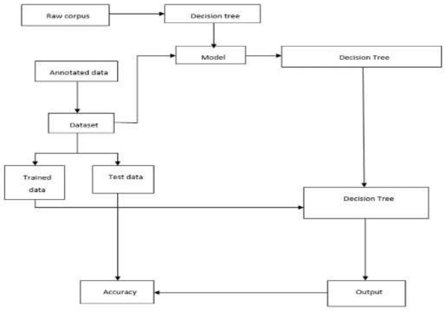


Figure: System Architecture

i) 3-Tier Architecture

The three-tier software architecture (a three-layer architecture) emerged in the 1990s to overcome the limitations of the two-tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two-tier architecture) by providing functions such as queuing, application execution, and database staging.

The three-tier architecture is used when an effective distributed client/server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. These characteristics have made three-layer architectures a popular choice for Internet applications and net-centric information systems.

ii) Modules

a) Dataset Collection

Dataset is taken from Kaggle website with has features as user id signup time purchase time purchase value device id source browser sex age ip address and label as fraud or not.

b) Preprocessing

In given data set many unwanted features are used which are device_id, source browser, user id are removed and time is converted to required format.

c) Split Dataset

In this stage data is collected from dataset and divided to testing and training and given input to algorithm and fit to algorithm.

V. SYSTEM IMPLEMENTATION

To conduct studies and analyses of an operational and technological nature, and to promote the exchange and development of methods and tools for operational analysis as applied to defines problems.

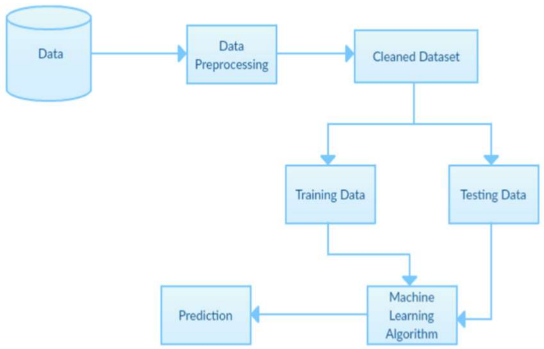


Figure: System Implementation

i) Logical design

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems design are included. Logical design includes ER Diagrams i.e. Entity Relationship Diagrams.

ii) Physical design

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified / authenticated, how it is processed, and how it is displayed as output.

iii) Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system.

iv) Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output.

a) Prediction Algorithm compare

Step 1 - Import the Libraries

We will start by importing the necessary libraries required to implement the RFC Algorithm in Python. We will import the numpy libraries for scientific calculation.

Step 2 - Fetch The Data

We will fetch the data from csv file using ‘pandas data reader’. We store this in a data frame ‘df’.

Step 3 - Split The Dataset

We will split the dataset into training dataset and test dataset. We will use 70% of our data to train and the rest 30% to test. To do this, we will create a split parameter which will divide the data frame in a 70-30 ratio.

b) Initialize RFC Model

After splitting the dataset into training and test dataset, we will instantiate Random Forest classifier and then fit the train data by using ‘fit’ function. Then, we will calculate the train and test accuracy by using ‘accuracy score’ function.

c) Prediction:

Based on given input values data is passed as array to trained model and prediction of fraud activity or not is detected.

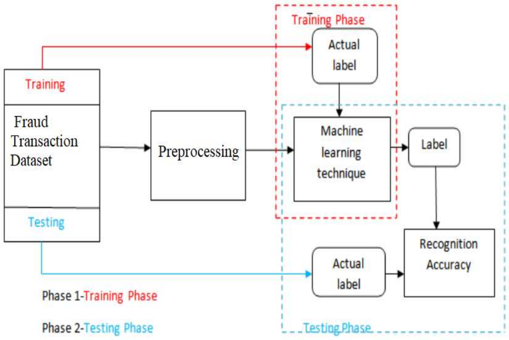


Figure: System Design

VI. ALGORITHM

a) Random Forest Algorithm

Step 1: In Random Forest, n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and Regression respectively.

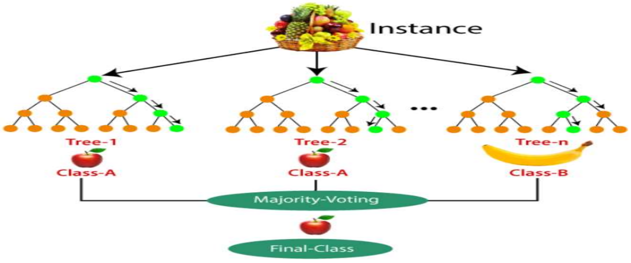


Figure: Random Forest Algorithm

For example: consider the fruit basket as the data as shown in the figure below. Now n number of samples are taken from the fruit basket and an individual decision tree is constructed for each sample. Each decision tree will generate an output as shown in the figure. The final output is considered based on majority voting. In the below figure you can see that the majority decision tree gives output as an apple when compared to a banana, so the final output is taken as an apple.

b) Important Features of Random Forest

1. Diversity- Not all attributes/variables/features are considered while making an individual tree, each tree is different.

2. Immune to the curse of dimensionality- Since each tree does not consider all the features, the feature space is reduced.

3. Parallelization-Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.

4. Train-Test split- In a random forest we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.

5. Stability- Stability arises because the result is based on majority voting/ averaging.

VII. RESULTS

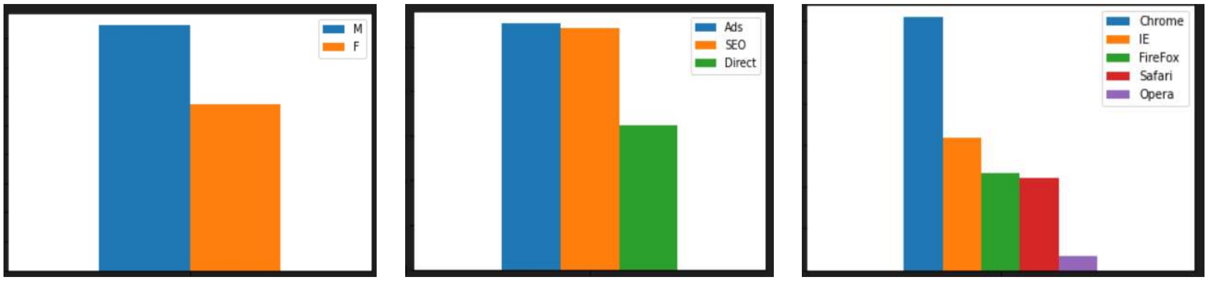


Figure: Graph

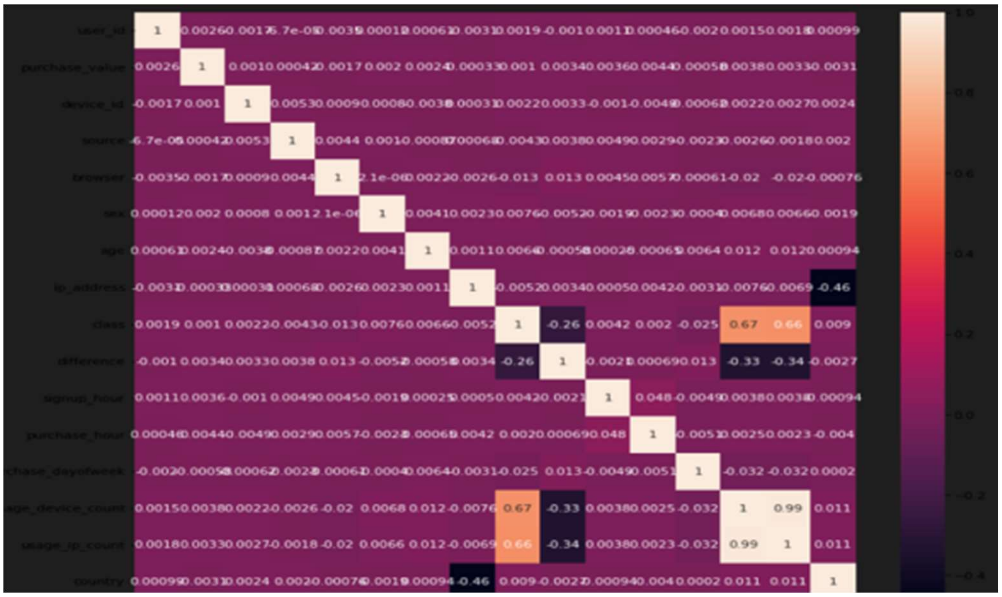
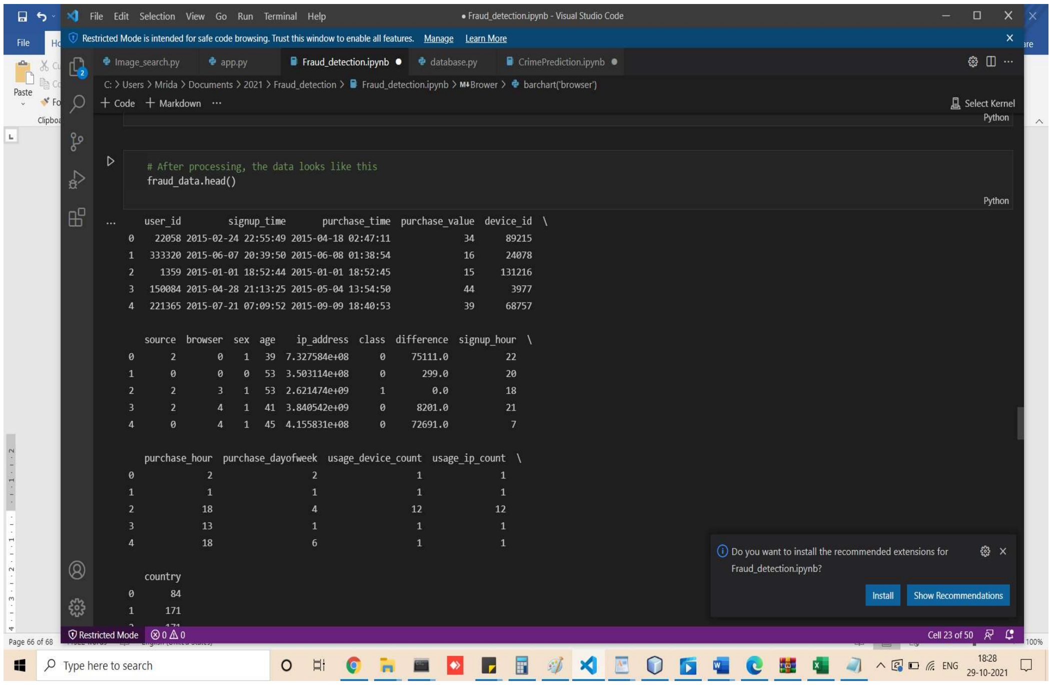


Figure: Analysis



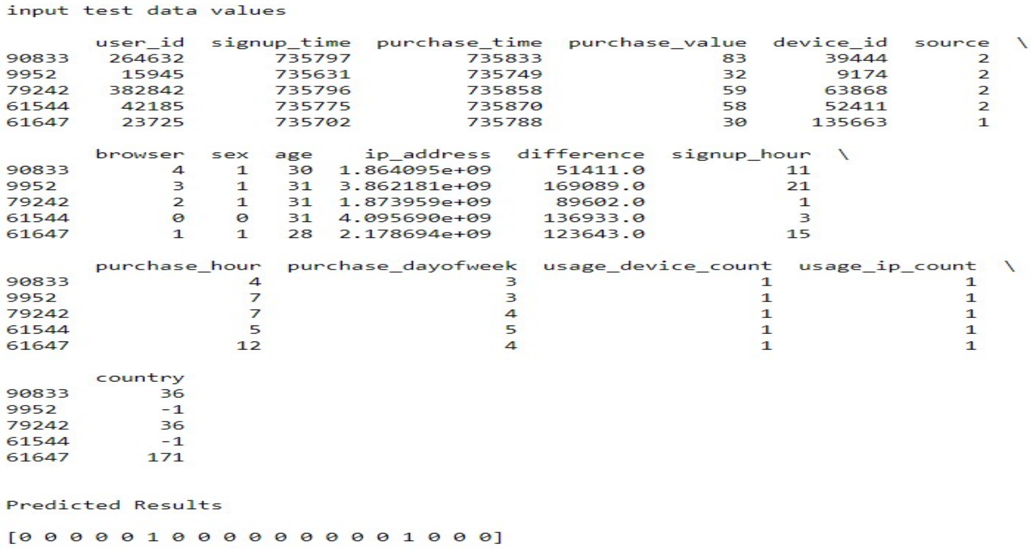
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

```
# After processing, the data looks like this
fraud_data.head()
```

```
   user_id         signup_time       purchase_time  purchase_value  device_id  \
0    22058 2015-02-24 22:55:49 2015-04-18 02:47:11              34      89215
1   333320 2015-06-07 20:39:50 2015-06-08 01:38:54              16      24078
2     1359 2015-01-01 18:52:44 2015-01-01 18:52:45              15     131216
3   150084 2015-04-28 21:13:25 2015-05-04 13:54:50              44       3977
4   221365 2015-07-21 07:09:52 2015-09-09 18:40:53              39      68757

   source  browser  sex  age    ip_address  class  difference  signup_hour  \
0       2        0    1   39  7.327584e+08      0     75111.0           22
1       0        0    0   53  3.503114e+08      0       299.0           20
2       2        3    1   53  2.621474e+09      1         0.0           18
3       2        4    1   41  3.840542e+09      0      8201.0           21
4       0        4    1   45  4.155831e+08      0     72691.0            7

   purchase_hour  purchase_dayofweek  usage_device_count  usage_ip_count  \
0              2                   2                   1               1
1              1                   1                   1               1
2             18                   4                  12              12
3             13                   1                   1               1
4             18                   6                   1               1

   country
0       84
1      171
```

Do you want to install the recommended extensions for Fraud_detection.ipynb?

Figure: Dataset



```
input test data values

        user_id  signup_time  purchase_time  purchase_value  device_id  source  \
90833    264632       735797         735833              83      39444       2
9952      15945       735631         735749              32       9174       2
79242    382842       735796         735858              59      63868       2
61544     42185       735775         735870              58      52411       2
61647     23725       735702         735788              30     135663       1

        browser  sex  age    ip_address  difference  signup_hour  \
90833         4    1   30  1.864095e+09     51411.0           11
9952          3    1   31  3.862181e+09    169089.0           21
79242         2    1   31  1.873959e+09     89602.0            1
61544         0    0   31  4.095690e+09    136933.0            3
61647         1    1   28  2.178694e+09    123643.0           15

        purchase_hour  purchase_dayofweek  usage_device_count  usage_ip_count  \
90833               4                   3                   1               1
9952                7                   3                   1               1
79242               7                   4                   1               1
61544               5                   5                   1               1
61647              12                   4                   1               1

        country
90833        36
9952         -1
79242        36
61544        -1
61647       171

Predicted Results

[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0]
```

Figure: Input and Prediction Results

VIII. CONCLUSION

This paper proposes a model based on information retrieval and Random Forest, Fraud Prevention Algorithm to proactively detect fraudulent merchants based on their past performance. In addition, the proposed model attempts to leverage the power of social media analytics to understand and con-sider what the society thinks about the services/products delivered by a particular seller through a marketplace. One key issue which is still open for discussion and hence available for future extensions on this model is the cold start problem. Random Forest, Fraud Prevention require input data for training the model and hence it is not practically possible to evaluate a new seller as past traits are not available in this case.

REFERENCES

- M.Chau and H. Chen, A Machine Learning approach Webpages filtering using content and structured Analysis, Decisio Support Systems, Vol. 44, No. 2, pp. 482-494, 2008.
- Daniel Zeng, Brian Kirkm Jeeny Stout , Crystal balls, Statistics, Big data and Psychohistory: Predictive Analytics and Beyond, IEEE Intelligent System, Vol. 30, No. 3, pp.114-122, 2015.
- Donald E. Brown, Ahmed Abbasi, Rayond Y.K. Lau, Predictive Analytics: Predictive modelling at the micro level, IEEE Intelligent System, Vol. 30, No. 3 , pp. 1541-1672,2015.
- Gao Huang, Shiji Song,Jatinder N.D. Gupta and Cheng Wu, SemiSupervised and Unsupervised Extreme Learning Machines, IEEE Transition on Cybernetics, Vol. 44, No. 12, pp. 2405-2417, 2014.
- Hau Hu, Yonggang Wen, Tat-Seng Chau and Xuelong Li, Toward Scalable Systems for Big data Analytics: A technology Tutorial, Vol. 2, No. 3, pp. 1556-1603, 2014.

- Hua Fang, Zhaoyang Zhang, Chanpaul Jin Wang, Daneshmand, Chonggang Wang, Honggang Wang, A survey of Big data research, IEEE Networking, Vol. 29, No. 5, pp.6-9,2015.
- M. R. Mesbahi, A. M. Rahmani, and M. Hosseinzadeh, "Reliability and high availability in cloud computing environments: a reference roadmap," Human-centric Computing and Information Sciences, vol. 8, p. 20, 2018.
- Alsmadi and H. Najadat, "Evaluating the change of software fault behavior with dataset attributes based on categorical correlation," Advances in Engineering Software, vol. 42, pp. 535- 546, 8// 2011.
- S. Chatterjee and A. Roy, "Web software fault prediction under fuzzy environment using MODULO-M multivariate overlapping fuzzy clustering algorithm and newly proposed revised prediction algorithm," Appl. Soft Comput., vol. 22, pp. 372-396, 2014.
- C. Jin and S.-W. Jin, "Prediction approach of software faultproneness based on hybrid artificial neural network and quantum particle swarm optimization," Applied Soft Computing, vol. 35, pp. 717-725, 10// 2015.
- P. J. García Nieto, E. García-Gonzalo, F. Sánchez Lasheras, and F. J. de Cos Juez, "Hybrid PSO–SVM-based method for forecasting of the remaining useful life for aircraft engines and evaluation of its reliability," Reliability Engineering & System Safety, vol. 138, pp. 219-231, 6// 2015.
- V. Balasubramanian, F. Zaman, M. Aloqaily, I. Al Ridhawi, Y. Jararweh, and H. B. Salameh, "A mobility management architecture for seamless delivery of 5G-IoT services," in ICC 2019-2019 IEEE International Conference on Communications (ICC), 2019, pp. 1-7.