DOI: 10.15439/2021R11
ISSN 2300-5963 ACSIS, Vol. 27

Artificial Intelligence applications in anomaly identification detection of big database

*

Phan Huy Thang
National Credit Information Center (CIC)
Hanoi, Vietnam
thangph@creditinfo.org.vn

Nguyen Thi Ngoc Anh
School of Applied Mathematics and Informatics
Hanoi University of Science and Technology
CMC institute of science and technology
Hanoi, Vietnam
anh.nguyenthingoc@hust.edu.vn,
ORCID 0000-0002-6555-9740

***Abstract*—Data matching is the process of finding, matching, and combining records from many databases or even within one database that belong to the same entities. All parts of the data matching process have been improved during the previous decade as a result of research in various disciplines such as applied statistics, data mining, machine learning, database administration, and digital libraries.Indeed, with the significant advance in artificial intelligence over the past decade, all aspects of the data identification process, especially on how to improve the accuracy of data matching. Firstly, this paper presents the process of comparing data, detailing the steps to perform pre-processing data, comparing the data fields of each record, classification, and quality assessment. Secondly, the paper introduces a method to expand the problem of identifying duplicate objects with big data. Third, the paper also provides specific aspects of unstructured data matching times. Moreover, the methodology of solving big data matching problems by machine learning is proposed. Finally, the proposed method is applied to the problem of database cleanup and identification of identifier abnormalities at the national credit centre CIC with correct results from 96% to 98%. The achieved results are not only theoretical but also practical in business operations at CIC.**

***Index Terms*—big data, abnormality detection, duplicate profiles, similarity, artificial intelligence**

I. Introduction

HUGE volumes of data are generated every day as a result of the ever-increasing sharing of information on the Internet. Big data really starts when we understand the value of the information hidden in the data. When we have enough resources, Machine Learning is the key to that technology. The relation between machine learning and big data has been developing thanks to the proliferation of data and vice versa. The value of big data depends on the data's ability to exploit knowledge. In the past 20 years, Bigdata and AI has strongly developed. The development of AI has come from three different factors: (1) advancement in the study of algorithms, (2) an increase in computing ability, and (3) an explosion of data.

Machine learning is capable of self-learning based on input data without programming specifically. The aim of machine learning is to make the system intelligent. Machine learning helps humans in processing huge amounts of information (Big data) that we face every day. Machine learning have three common types: supervised, unsupervised and semi-supervised.

However, data labelling for big data is expensive and takes time consuming, most huge datasets are unlabelled and contain a variety of attributes from various types of data [10], and as a result, unsupervised learning on big data has recently resurfaced. Furthermore, the scale of data has a substantial impact on the complexity of unsupervised machine learning algorithms, making real-time unsupervised learning on huge data a serious challenge. Clustering problem is one of the most often used unsupervised learning strategies in data mining. It enables analysts to automatically generate groups in datasets based on the similarity records, which can be of many types such as texts, numbers, category, or a combination of these types [10]. Cluster analysis is widely employed in a variety of domains, including economics, science and technology.

Currently, the research topic of detecting abnormality based on AI in processing big data has not been officially published in scientific papers in Vietnam. The practical application of AI has not been studied and applied to abnormality detection.

Until this time, other publications are mainly about mathematics tools, assessing the potential of big data analytics with businesses, applications in the field of genetic classification, assessing customer behavior in the purchasing process, Image Processing, Language, and Speech Processing. Many researchers have paid attention to the solution of detecting duplicate records (based on similarity) in big database and merging them into a unique identifier [1] [3] [4] [6]. Detecting duplicate records is meaningful in:

* Searching and removing duplicate records in the entire

The authors group would like to thank the National Credit Information Center (CIC) and, which cooperated successfully in applying machine learning technology to the actual problem of CIC.

database;

* Merging the data from other sources with an internal database;

* Improving data quality and consistency;

* Creating more adequate, in-depth, and comprehensive data tables;

* Data cleaning

In recent years, machine learning algorithms for record duplicate detection have been studied widely. Some machine learning methods such as K-means, Adaptive neuro-fuzzy inference system (ANFIS) [2], Support Vector Machine, Naïve Bayes, Decision Tree [5], DSC++ algorithm (Arfa Skandar et al. [8]), Semantic - Syntactic Method (Djulaga Hadzic and Nermin Sarajlic [9]). This paper aims to detect duplicate records in the entire database. When the new code calculates the new record duplicate similarity compared with the existing records in the database, duplicated records are matched to a unique identifier record.

The paper's layout consists of 4 sections: Section 1 -An overview of cleaning up the database by finding similar identifiers. Section 2 shows the methodology of big data processing application to solve the problem with large computational complexity; Section 3 - Practical applications at CIC; Section 4 - conclusions and discussion.

II. Methodology

A. Processing the data for the similarity records in the general model

The general model of data processing is done in 4 steps, the illustration below:

Step 1: Data collection: The data were collected from CIC's databases and will be stored for further processing steps. Data includes many different types: numerical data, categorical data, or text data. The data on detected duplicate records will be used to evaluate the next steps.

Step 2: Data processing: This is an important step, as a premise for the next processing steps. In this step, there will be different processing methods that depend on the type of data. Then, the importance of collecting data is evaluated. After that, important fields of data are extracted for developing a model.

Step 3: Data exploration: The data will be divided into two parts of the training dataset and test dataset in the previous step, they will be used to develop the model. Data will be assessed according to two methods:

+ Similarity detection has based on the rules that CIC developed, using similar measurement methods.
+ Similarity detection has based on machine learning models. The model is continuously trained based on historical data (instances in which duplicate records have been detected) and user reviews to make the model more and more optimal.

Step 4: Evaluation: New profiles need to be assessed through an established model to give similarity evaluation scores. The higher the score of these records is, the greater the similarity is, and vice versa.

B. Locality sensitive hashing

Locality sensitive hashing (LSH) is a probabilistic algorithm that hashes similar input items into the same buckets. Data-independent hashing approaches, such as locality-sensitive hashing, are commonly used in hashing-based approximate closest neighbour search algorithms. This algorithm solves the problem is that given a large number N in the millions or even billions of text documents, find pairs that are near duplicates document of text. The main LSH $Document_i, Document_j$ in dataset are considered:

A min hash function $hast$ such that:

- if $Similarity(Document_i, Document_j)$ is high, then with high probability $hash(Document_i) = hash(Document_j)$
- if $Similarity(Document_i, Document_j)$ is low, then with high probability $hash(Document_i) \neq hash(Document_j)$.

Hashing documents into buckets, pairs of near duplicate documents hash into the same bucket. This is the key method to solve the big dataset that see Figure 2.

C. Split records

In the problem of data reduction and matching big databases, the complexity of the algorithm is the factor considered at first. This helps ensure that the solution processing times are within an acceptable threshold. With similarity matching algorithms for data reduction, when performing a comparison on a database consisting of n records, the complexity is $O(n^2)$. However, it is more optimal to perform a similarity comparison throughout this database when comparing only a record with records that are likely to be similar to this record. A split record is used for this purpose.

Splitting records is clustering the database into small clusters (called bucket). With the characteristics of the problem of removing duplicate records and having the feature of finding similar records, these buckets must fulfill similarity requirements as follows:

$$\begin{cases} sim(r_i, r_j) \geq t & \forall r_i, r_j \in bucket_k \\ sim(r_i, r_j) \leq t & \forall r_i \in bucket_m, r_j \in bucket_n, m \neq n \end{cases} \tag{1}$$

Where:

$sim(r_i, r_j)$: the degree of similarity between 2 records r_i and r_j.

t: is a given similarity threshold.

$bucket_k$: the cluster of records k.

From the above requirements, the records after being classified into small clusters must fulfill: records in the same cluster must have similarity level above a given threshold, records in other clusters must have similarity less than the threshold given. There are many algorithms used for database clustering purposes, which have based on the used type of data. The

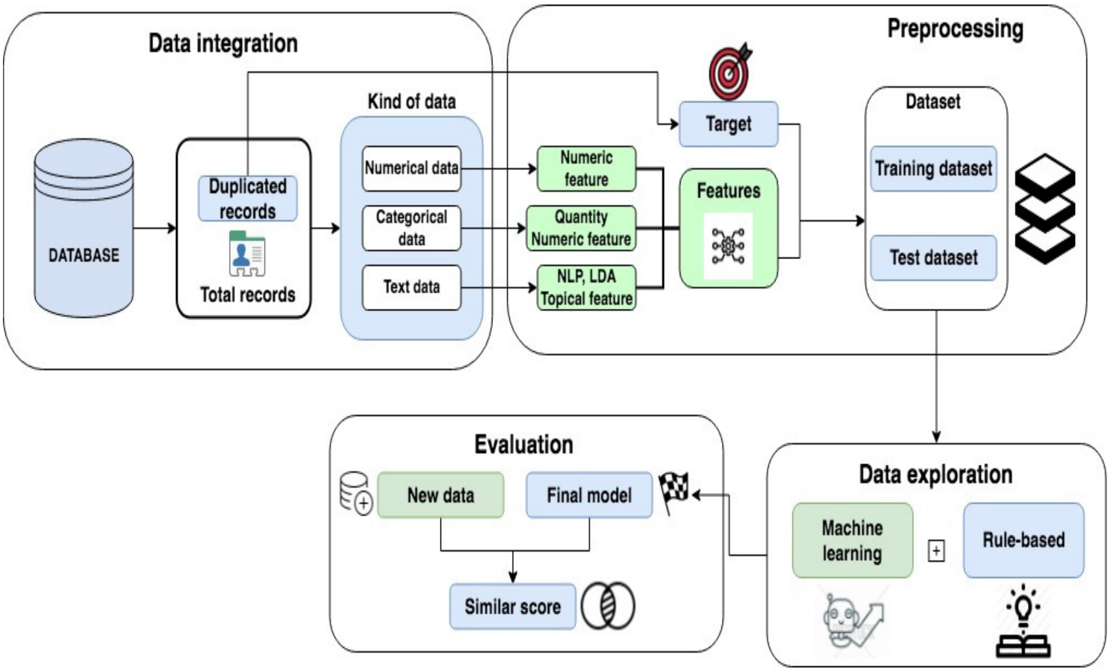

Fig. 1. General model of data processing for similar records.

algorithms are into two groups as follows: Group of Algorithms used for numerical data: K-mean Clustering, DBScan, PAM,... Group of Algorithms used for string data: Multibit Tree, Succinct Multibit Tree. In this problem, identity data including string data (such as Name, ID Number) and numeric data (such as Date of Birth, Address), we have combined both two groups of algorithms for the purpose of splitting records.

D. Similarity

The similarity is a numerical measure of how different two or more data objects are. This measure is used to reflect the intensity of the relationship between data objects. The problem of similarity calculation is illustrated as follows: Two attribute values are d_i and d_j. The aim is to find a value $S(d_i, d_j), S \in [0; 1]$, that shows the similarity between d_i and d_j. The higher the value is, the greater the similarity between d_i and d_j. The similarity value is used in some functions: Jaccard index, Jaro distance, Haversine formula for points.

1) Jaccard index: The Jaccard index measures similarity between sample sets based on statistical methods. Accordingly, the similarity value between two A and B strings can be calculated as:

$$sim(A, B) = \frac{A \cap B}{A \cup B}, \qquad 0 \leq sim(A, B) \leq 1 \tag{2}$$

2) Jaro distance: The Jaro distance defines the measure of similarity between two strings. Given two s_1 and s_2 string, the Jaro distance d between two strings is defined as:

$$d = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) \tag{3}$$

Where: m: the number of matching characters between two strings; t: $\frac{1}{2}$ of the number of transposition steps (transposition). This function counts the number of characters that are common in two strings and the distance is not greater than the following value: $\frac{\max(|s_1|,|s_2|)}{2} - 1$. Transpositions are defined as the number of common characters in two strings (but the order in the sequence is different) divided by 2.

3) Haversine distance: Assuming P_1 and P_2 points are converted into the corresponding latitude and longitude $long_1$, lat_1 and $long_2$, lat_2. The Haversine distance d is calculated as:

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_1).\cos(lat_2).\sin^2\left(\frac{\Delta long}{2}\right) \tag{4}$$

$$c = 2.\arctan 2(\sqrt{a}, \sqrt{1-a}); d = R.c \tag{5}$$

Where: $\Delta lat = lat_2 - lat_1$; $\Delta long = long_2 - long_1$; R: the radius of the Earth (R=6731km).

Finally, in order to calculate the similarity between two points, it is possible to use the Haversine distance to a value in the segment $[0, 1]$ by using functions such as sigmoid, exponential function,...

E. Support Vector Machine method

Support Vector Machine (SVM) is a statistical and computer science concept for a set of associated supervised learning methods for classification and regression analysis. The standard SVM multiply inputs and classifies them into two different classes. Therefore, SVM is a binary classification algorithm. SVM was developed by Vapnik, and his colleagues in the 1970s in Russia then became popular and famous in the 1990s. SVM has many practical applications such as genetic analysis, marketing, or facial recognition [7].

Support Vector Machine (SVM) is a Supervised Learning algorithm that has been used to divide data (Classification) into separate groups. A Support Vector Machine builds a hyperplane or a set of hyperplanes in a multidimensional or infinite-dimensional space, which can be used for classification, regression, or other tasks. For the best classification, the hyperplanes are intuitively located as far away from the data points of all layers (called margins) as possible. In general, the larger the margin is, the smaller the generalization error of the classification algorithm is.

With the binary classification problem, the SVM constructs a separating hyperplane located as far away from the data points of all layers (called margin) as possible. Assuming that there is training dataset D with n points, each point x_i belongs to one of two class labels: $y_i : D = \{(x_i, y_i)\forall x_i \in \mathcal{R}, y_i \in \{-1, 1\}\}$. Each hyperplane is described by the equation and to maximize the margin with the following requirements:

- $wx_i - b \leq 1$ if the data point belongs to the positive class.
- $wx_i - b \geq 1$ for other classes.
- Or $y_i(wx_i - b) \geq 1 \quad \forall 1 \leq i \leq n$

The optimal problem for SVM can follow quadratic programming formulation:

$$\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j x_i^t x_j \tag{6}$$

Subject to:

$$\begin{cases} \lambda_i \geq 0 \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

In the case of data points that are non-linear, a soft margin SVM or a non-linear function can be used to map the input data space to a larger number space that this space can split with a hyperplane. This technique is called a kernel trick.

F. Proposed model

The proposed model is shown below:

The original raw data are available in database A and Database B. In the case of the data reduction problem, database A and database B represent a set of data. In the matching problem, database A and database B are two different data sets. Implementation steps include:

- *Step 1*: Data pre-processing: In this step, the data fields in both databases A and database B are normalized to a common format, such as the names written in the right order of Surname – Middle name - Name, or the date of birth in the format of DD-MM-YYYY,...
- *Step 2*: Indexing: The normalized data will be put into the indexing system, and then the system calculates the records based on the data of the record. The indexing algorithm is selected to assure that adjacent records also have similar data content.
- *Step 3*: Creating suspected pairs of similarity: Based on indexing the records which are generated from step 2, the system collects records into the bucket (containing adjacent indexing records) and creates suspected pairs of similarity according to the method of exhaustion.
- *Step 4*: Comparing similarities: To compare the similarity on the data content of the record pairs which have similar doubt. The rules for evaluating compared information can be used. For example, the rule of matching the records have entire common names, dates of birth, ID number with a 1-character difference. Besides, the machine learning model is also applied for calculating similarity. After comparing the records, they are grouped into three categories: records with a high similarity, records with disparity, and records with suspected similarity (corresponding to the achieved scores). Similar records should be checked by CIC's specialized division.
- *Step 5*: Evaluating: After pairs of similar suspected records are directly checked about similarity ability, the system will record labels of pairs of records whether they are in common or not. Since it provides a review of the effectiveness of the solution.
- *Step 6*: Training the machine learning model for similarity scoring: Based on the user rating on pairs of duplicated suspected records, the system uses these ratings as labels as input to the model training process. This training-evaluation process will be run in a loop. It helps this model improve accuracy over time.

TABLE I
TABLE 1: DESCRIPTION OF DATA

Column name	Description	Data types
TENKH	Full name	Varchar2(150)
SOCMT	Identity card number	Varchar2(30)
ADDRESS	Address	Varchar2(250)
GIOITINH	Gender	Varchar(1)
DTHOAI	Phone number	Varchar(50)
MSTHUE	Tax code	Varchar(30)

III. APPLICATION AND RESULTS

Data description: The data is extracted from CIC's customer database, that will be used to perform the duplicate record detection problem. The information of customers used for the problem includes Full name, ID number, Address, Gender, Phone number, Tax code. The data are described in Table 1.

Result In this paper, with the experimental scale, we have used a data volume of 10 million records with addresses in the North of Vietnam. In addition to the current data of the credit data system, we also include a data set of records that have been assessed by the specialized divisions at CIC as duplicates (5,000 records).

This data set will be included in the duplicate detection tool given by the authors in the proposed model and evaluated

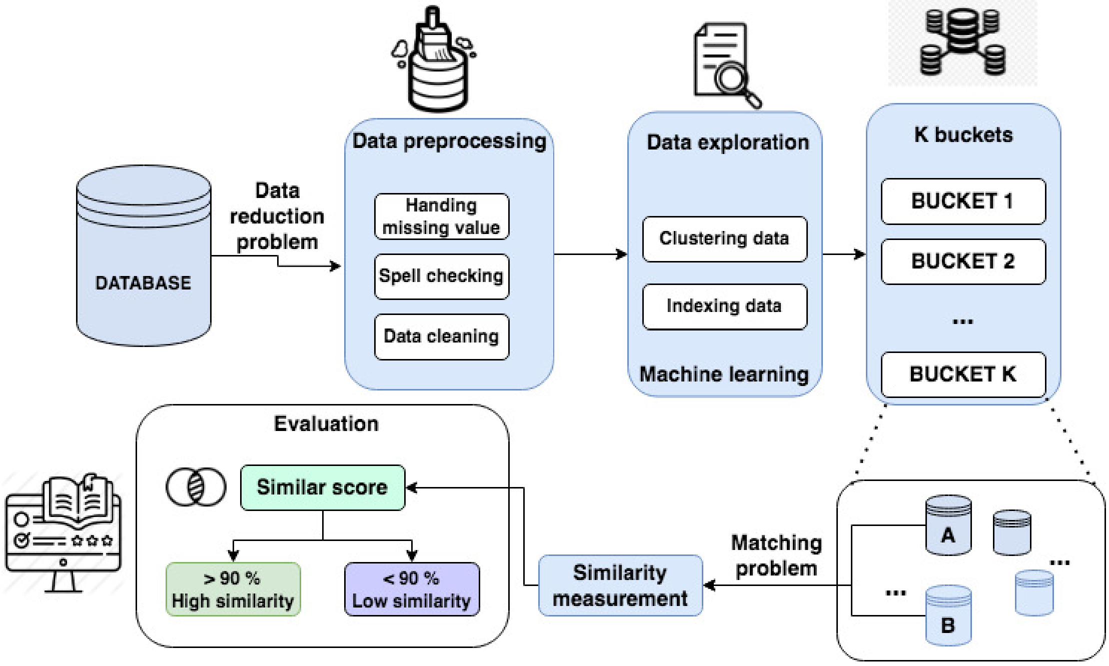

Fig. 2. General model of data processing for similar records.

TABLE II
NUMBER OF SUSPECTED DUPLICATE RECORDS DETECTED

Similarity threshold	Number of suspicious records
0.98	7,800)
0.95	9,400
0.90	15,200

through indicators including the number of suspicious records detected and the coverage rate with duplicate labeled records.

After two hours blocking process, the solution return a data set of 10 million records to 800,000 blocks (approximately 12 times), each block contains about 12 records on average, thereby reducing the number of comparisons needed from $10,000,000 \times (10,000,000 - 1)/2$ 50×10^{12} to $800,000 \times (12 \times 11)/2$ 52×10^{6} nearly 1 million times.

The similarity is calculated on the records in the same data block in 3 hours to evaluate the similarity of the records and the efficiency through the number of detected similar pairs of records when the similarity exceeds the permitted threshold, the results are as Table 3. The records have been labeled similarly by specialized division at CIC, the authors also evaluated the coverage of the solution with this data group, and the specific results are shown in the Table 3. The quality of the machine learning model for duplicate record detection was also assessed after training the model with 80% of the labeled records (4000 pairs of records with duplicate labels) on a test dataset of 1000 pair of records, for the specific result see Table 4.

TABLE III
COVERAGE RATES WITH DUPLICATE DATA

Similarity threshold	Coverage rate
0.98	73%
0.95	81%
0.90	97%

TABLE IV
EVALUATION METRIC OF MACHINE LEARNING MODEL FOR PROPOSED

Metric	Result
Accuracy	0.97
Precision	0.96
Recall	0.98
F1 score	0.97

IV. CONCLUSION AND DISCUSSION

Duplicate Record Identification is a solution to detect duplicate records (based on similarity) in a big database system and combine them into a unique identifier.

Detecting and removing duplicates helps search and remove duplicate records in one or more databases. Identifying duplicate records contributes to data cleaning, and cleaning usually takes up to 80% of the data-maker's time to use. The solution we propose will help reduce at least 40% of the time for this step.

Duplicate detection and elimination contribute to data preparation for in-depth intelligence analysis. The quality of in-depth intelligence analysis reports depends entirely on the reliability of the data source. Eliminating duplicate data is a prerequisite in building highly reliable data sources.

In particular, the identification of duplicate records contributes to automatic fraud prevention. This work allows companies to trace the steps, enabling investigations to arrive at the source of the problem.

The paper points out the similarity assessment by calculating similarity scores to assess the degree of duplication between records.

The proposed methods were applied by the authors to the problem of the National Credit Information Center CIC and gave correct results from 96% to 98%.

Acknowledgements The author's group would like to thank the National Credit Information Center (CIC) and, which cooperated successfully in applying machine learning technology to the actual problem of CIC.

References

[1] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, Vassilios S. Verykios, Duplicate record detection: A Survey, In: IEEE Transactions on knowledge and data engineering 2007, Vol.19.

[2] G. Ranganathan, V.Bindhu,. Jenifer Raj, Duplicate record detection using intelligent approaches, In: International Journal of Pure and Applied Mathematics 2018, Vol.119, No.12, pp.13077–13087.

[3] Peter Christen, Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection; Springer (2012).

[4] Batini, C., Scannapieco, and M.: Data quality: Concepts, methodologies and techniques. Data-Centric Systems and Applications. Springer (2006)

[5] Arasu, A., Götz, M., Kaushik el at: On active learning of record matching packages.In: ACM SIGMOD, pp.783—794. Indianapolis (2010).

[6] Alvarez, R., Jonas, J., Winkler, W., Wright, R .: Interstate voter registration database matching: the Oregon-Washington 2008 pilot project. In: Workshop on Trustworthy Elections, pp.17—17. USENIX Association (2009).

[7] Roya Hassanian-esfahani, Mohammad-javad Kargar , Sectional MinHash for near-duplicate detection, In: Expert Systems with Applications, Volume 99, 1 June 2018, pp.203–212.

[8] Arfa Skandar, Mariam Rehman,Maria Anjum, An Efficient Duplication Record Detection Algorithm for Data Cleansing, In: International Journal of Computer Applications, Volume 127, October 2015, pp.28-37.

[9] Djulaga Hadzic and Nermin Sarajlic, Methodology for fuzzy duplicate record identification based on the semantic-syntactic information of similarity, In Journal of King Saud University - Computer and Information Sciences, Volume 32, 2020, pp.126-136.

[10] Toan Nguyen Mau and Van-Nam Huynh, An LSH-based k-representatives clustering method for large categorical data, Neurocomputing, volume 463, pages 29-44, year 2021,