Computer Assisted Methods in Engineering and Science
30(1): 55–71, 2023, doi: 10.24423/cames.472

This article belongs to the *Special Issue on Information Theory with Artificial Intelligence in 5G/6G Networks* edited by Dr. A.K. Budati, Dr. M.K. Hasan, Dr. R. Arthi, Dr. K. Thangavel

Machine Learning-Based Business Rule Engine Data Transformation over High-Speed Networks

K. NEELIMA*, S. VASUNDRA

Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Anantapur, Ananthapuramu, Andhra Pradesh, India;
e-mail: vasundras.cse@jntua.ac.in

* *Corresponding Author e-mail: neelimakenpi@gmail.com*

Raw data processing is a key business operation. Business-specific rules determine how the raw data should be transformed into business-required formats. When source data continuously changes its formats and has keying errors and invalid data, then the effectiveness of the data transformation is a big challenge. The conventional data extraction and transformation technique produces a delay in handling such data because of continuous fluctuations in data formats and requires continuous development of a business rule engine. The best business rule engines require near real-time detection of business rule and data transformation mechanisms utilizing machine learning classification models. Since data is combined from numerous sources and older systems, it is challenging to categorize and cluster the data and apply suitable business rules to turn raw data into the business-required format. This paper proposes a methodology for designing ensemble machine learning techniques and approaches for classifying and segmenting registered numbers of registered title records to choose the most suitable business rule that can convert the registered number into the format the business expects, allowing businesses to provide customers with the most recent data in less time. This study evaluates the suggested model by gathering sample data and analyzing classification machine learning (ML) models to determine the relevant business rule. Experimentation employed Python, R, SQL stored procedures, Impala scripts, and Datameer tools.

Keywords: CRISP, DM, data mining algorithms, business rules, prediction, classification, machine learning, deep learning, AI design, method.

1. Introduction

In today's business world, the data keeps changing and it requires continuous effort to identify and analyze the new patterns of the data received. The new patterns always demand deep analysis, new business rules and logics to transform

the raw data into business-required formats. Incorporating this requirement is very time-consuming as it involves various phases such as analysis, identifying business rule, developing the rule in extract transform tool (ETL) or any other deployments, leading to delay in the availability of the data to the end user. In this competitive environment, there is the need for the data to be available to the user at the earliest. Also, if the new patterns are not identified at the beginning, then there is a good chance that data get into database in incorrect formats, which need to be corrected later. Hence, it is required to distinguish known patterns, partially known patterns and unknown patterns at the beginning so that the unknown or partially known patterns do not get updated in the database with incorrect business rules and wrong formats. This challenge can be addressed with data mining and deep learning techniques that give a capacity to automate the various phases of handling the new patterns in the best-fitted manner. Although there already exists some research done on real-time ETL processing, there are still areas that need revisiting to make a sophisticated and robust system. In the past, partial and minimal research was conducted on automating the rule engine for data integration with minimum manual interference. Hence, real-time data integration is still an open problem and has recently gained popularity. Near real-time data integration and/or no manual intervention is required for deriving the formatting rules. The business rules should be mapped and created automatically by identifying the data anomalies and issues to handle the business decisions effectively. Rule engine mining using deep learning is required to identify the best-fit transformation rule for complex data. Also, machine learning models will have to be used to identify the right data categories and outliers. The rule engine should be able to collect the raw and formatted historical data and frame a data mapping model by applying deep learning methodologies to the obtained historical data.

Any system that leverages human capabilities of interaction, learning and insight at a level of complexity that ultimately can override our own abilities can be interpreted as artificial intelligence (AI). Machine learning (ML) can be deciphered as a subset of AI that involves programming systems to perform a specific task by learning without having to code every rule-based instruction. Deep learning (DL) is a subset of ML where systems can learn hidden patterns from the data, combine them together, and build much more effective and efficient decision rules. DL is based on artificial neural networks (NNs). Utilizing these maturing and advancing technologies in the area of data integration can lead to building an automatic business rule engine that provides faster availability of transformed data to the end-users and reduces human efforts. Based on the complexity of the data patterns, either ML or DL can be chosen to identify erroneous or eccentric data and classify the data into buckets of known, partially known, or completely unknown patterns.

Applying these technologies not only helps to spot incorrect data and outliers at the beginning but also helps in identifying the best-fit business rule to be applied in order to automatically transform the data into the business required format. Also, the model can learn and correct itself each time it improves the learning rate and counters the error rate. The title insurance companies maintain the data of the county offices recorded information called document registers. The registered number is the unique representation of the recording of a registered document at registration/revenue/court offices. The registered documents are identified uniquely by a document number on them. Hence, it is one of the very critical items to deal with for businesses maintaining similar information. The documentation of this critical field is maintained differently based on the regions and keeps changing constantly with time for various reasons. Thus, the data handling of such constantly changing field has become a great challenge and the need for automatic identification of relevant business rule and transformation by data mining using machine learning has become pressing priority.

2. Literature review

Contreras-Ochando *et al.* [1] have proposed ways to reduce the background knowledge primitives by selecting either domain data or ordering the appropriate primitives. Kratsch *et al.* [2] performed a comparison of DL and ML techniques and found that DL methods were far better when compared with ML methods. The key parameters such as accuracy, F-score, variant-to-instance ratio and others were found to be better using DL methods when compared with ML techniques. In the model proposed in this present work. some of these metrics categories were applied for the model's performance evaluation. Guha and Samanta [3] thoroughly analyzed the hybrid model for anomaly detection by using multiple classes as positive classes for the title domain as there exists high dimensionality in the text data. Sun *et al.* [4] proved that ML methods pose challenges; therefore, optimization in every step is required while constructing and applying such models. Nagar and Singh [5] provided a detailed review of some of the ML algorithms and applications of those algorithms. Ndirangu *et al.* [6] confirmed that ensemble and heterogeneous models provide optimal results when performing classification of multi classes. A combination of random forest, AdaBoost algorithm and others were used in their proposed ensemble method and 10-fold stratified cross-validation was performed as well. Wahid *et al.* [7] proposed a simple approach using entity embedding techniques for categorizing multi-dimensional data. Makani and Reddy [8] summarized some of the salient features and notable constraints for selecting the right approach, which were the key inputs for improving the proposed model.

3. Data preprocessing and transformation

To evaluate the proposed dynamic data transformation model, one of the critical fields of land registered documents, which is document number is selected. This model is built by using historical data of the transformed information along with the raw data source. Since each registered number is unique in nature and cannot draw a common business rule by considering the registered number as is, it is required to prepare the data to a standard format before the business rules for data transformation are extracted. Processing and transformation to a standard format are done as per the business requirements. Firstly, the sample data from the historical database is considered based on the associated columns on which the business rule depends. Associated columns can be obtained by domain knowledge or by deriving correlations between the columns and/or by applying principal component analysis (PCA). For the title insurance domain, the registered numbers are very much associated with the region they are registered in, and the year they are recorded. Hence, samples of registered numbers are drawn based on the regions covering all the years from the historical data.

For example, records are selected from each of the region for one or two random records per month. Preprocessing of the data plays a significant role as it helps to normalize and brings the varied data into a standard format that can help build a common business rule engine. During this step, the raw data, which is the same format as it appears on the registered document image, is converted to a pattern by replacing all the numbers, letters, and special characters with a respective character. Minimum analysis on the raw data must be conducted and basic knowledge of what the field constitutes and what kind of information it contains must be obtained. For example, the registered number generally is represented by a serial number and/or contains a date part based on the region in which it is recorded. In few regions, some special characters and letters are also included in the registered number. Based on the elements of the registered number contains, it is brought up to a standard format by replacing these elements with a pattern.

In this case, it is assumed that a registered number can possibly have a date part, a serial number part with/without leading zeroes, and an extension part that is usually a letter and/or special character. To format data, many techniques can be used to take the required features, programming features of Python/R for handling the text, numbers, and regular expressions. The formatting can also be done using structured queries using any of the database systems such as Oracle, SQL, MariaDB, MongoDB and others.

For example, if the source registered number is 2019004569, it is assumed that the year, in which the registered document is recorded is the first part; hence, a pattern CCYY is chosen to replace the same and the rest is a serial number with

leading zeroes. Leading zeroes are represented with the character "Z" and numbers are replaced with "N". Hence, the source can be brought into a standardized pattern as CCYYZZNNNN. Similarly, 116022807 to NNNNNNNNN, 20-PB5678 as YY-AANNNN and so on, as depicted in Table 1. The same patterning of the target number is also done.

Table 1. Input registered document transferred to the pattern system.

Region	Source_Pattern	Source_Raw_Data
REG1	CCYYZZZNNNN	20190005678
CAP2	NN-NNNNNNN	66-9300949
BEVN	CCYYZZNNNNN	20160056874
SHEL	YY-ZZZNNNNA	17-0004567B
KWIT	NNNNNNN-YY	8545912-19
BITS	CCYYMMDD-NNN	20140301-569
REG8	NN-NNNN-NN	55-4510-85
NMPS	NNNNNNNN-CCYY	84569920-2002

4. Data mapping model

A data mapping model can be used to map the obtained source and target pattern with data transformation dictionary created using historical data. After patterning both source and target, the most frequent source and target patterns are chosen for each associated column to be part of the business rule engine dictionary (BRED). A threshold value can be defined either by a fixed number or based on the number of documents in that region to qualify those frequent patterns having a bigger count the threshold value to be part of BRED. When the new input registered numbers are received then, they are converted to the

Table 2. Input and output registered document transformed into the pattern system.

Region	Source_Pattern	Target_Pattern	Source_Example	Target_Example
REG1	CCYYZZZNNNN	CCYY NNNN	20190005678	2019 5678
CAP2	NN-NNNNNNN	NNNNNNNNN	66-9300949	669300949
BEVN	CCYYZZNNNNN	NNNNN	20160056874	56874
SHEL	YY-ZZZNNNNA	CCYY NNNNA	17-0004567B	2017 4567B
KWIT	NNNNNNN-YY	CCYYNNNNNNN	8545912-19	20198545912
BITS	CCYYMMDD-NNN	CCYYMM NNN	20140301-569	201403 569
REG8	NN-NNNN-NN	NNNNN	55-4510-85	451085
NMPS	NNNNNNNN-CCYY	YYNNNNNNNN	84569920-2002	0284569920

standard pattern. The data mapping is done using associated attributes and source pattern against the BRED to obtain the respective target pattern. Source raw data can then be formatted as target data using data indexing logic.

The data transformation steps are given below:

Step 1: Get the sample based on associated attributes from historical data.

Step 2: Standardize source and target element by replacing them with a pattern.

Step 3: Identify the most frequent patterns by grouping them on associated columns and consider the patterns with the count having the least threshold count to be part of BRED.

Step 4: New source data for which the target format should be identified and formatted accordingly is brought to standardized format by patterning.

Step 5: Source data pattern along with associated attributes is mapped against BRED to obtain target pattern.

Step 6: Data mapping to the index data model: index or position of each source byte from which the target data is formatted is obtained in target format from BRED.

Step 7: Apply automatic data transformation logic using the obtained index data model.

5. Data conformance using AI and ML

Data conformance is an important aspect of various domains and title insurance is no exception. Data conformance covers validity, accuracy, consistency, integrity, timeliness, and completeness of the data. The clustering mechanism in ML and the supporting techniques can be used to convert the manual data collecting to automatic. Applying this beforehand helps in the identification of outliers [9], completeness of the data and any irregularities before the data integration or transformation is initiated against the data irregularities being identified after the data advances to the system crossing all the stages in the current traditional methods. This also helps in identifying if the data is labeled or unlabeled or semi labeled based on which best suit classification model can be applied to the respective grouped data.

With increasing data dimensionality and in composite population scenarios, the complexity of detecting anomalies and error rate increases. Automated data conformance is the least explored. So, in the proposed ensemble technique, K-means clustering and deep embedded clustering mechanism are applied and evaluated to address data conformance in this domain, which successfully identifies about 99% of the irregularities. Figure 1 depicts the K-means cluster plot with three clusters, each in distinct color representing different recommended pattern by the model and depicting an outlier too.

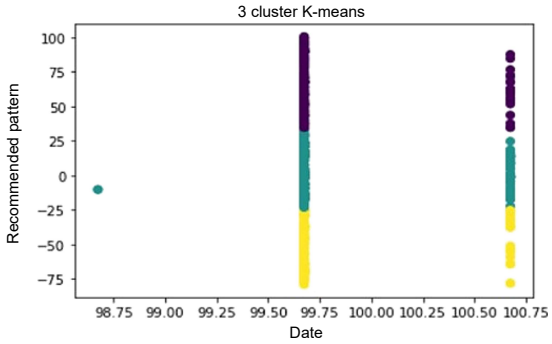


FIG. 1. Three cluster K-means depicting outlier data.

6. Design methodology

Cross industry standard process for data mining (CRISP-DM) methodology, depicted in Fig. 2, is applied and adopted for data collection, data profiling, exploratory data analysis, data mining, model evaluation and deployment for the proposed work. The evaluation approach and the metrics such as precision and recall can be taken later for further conformance of the results. It is important to formulate the right set of clusters by repeated assimilation of regressed data. The method can be developed to particularize the algorithm for detecting a specific anomaly from the given data set. Inductive learning methods can be applied for setting up the rules for deductive data correction. Some of the ensemble algorithms can be used further for evaluating the data quality and prediction models. Compared to the other two methods of KDD and SEMMA, CRISP-DM is more adaptable and can be applied to real-time systems in a phased manner.

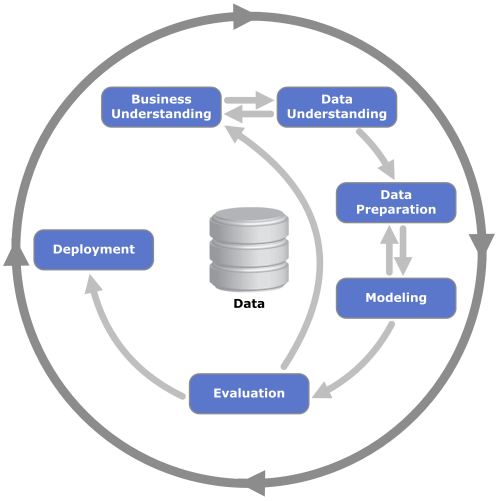


FIG. 2. Crisp-DM methodology (Source: Kenneth Jensen/Wikimedia Commons).

Figure 3 represents hybrid evaluation framework for effective business rules identification and classification using ML techniques. In the first layer, data conformance using AI and ML algorithms is applied to identify the data validity, outliers, missing information, and data completeness. Validated data is then clustered using K-means or deep embedded clustering methods. Formed clusters can be mapped to the existing data mapping model in order to identify if the cluster is labeled or unlabeled data. Labeled data is then classified using a supervised algorithm for which the source patterns are known and trained. For the partially known or unknown labels, data can be classified using unsupervised classification algorithms or DL NN algorithms. Some classification algorithms such as KNN, naive Bayes, and decision tree are to be applied to classify the output class label to which the formatted pattern can be closely fit. For the unknown variables or features, considering the relative attributes such as region or registered document type and others, a NN model is built to classify the input to output formatted pattern. Applying a NN-based reinforcement and rewarding system to this model are necessary. The efficacy of the automatic classification model is then calculated. The model is designed to be able to learn new patterns when received and, to unlearn the incorrectly identified formatted patterns.

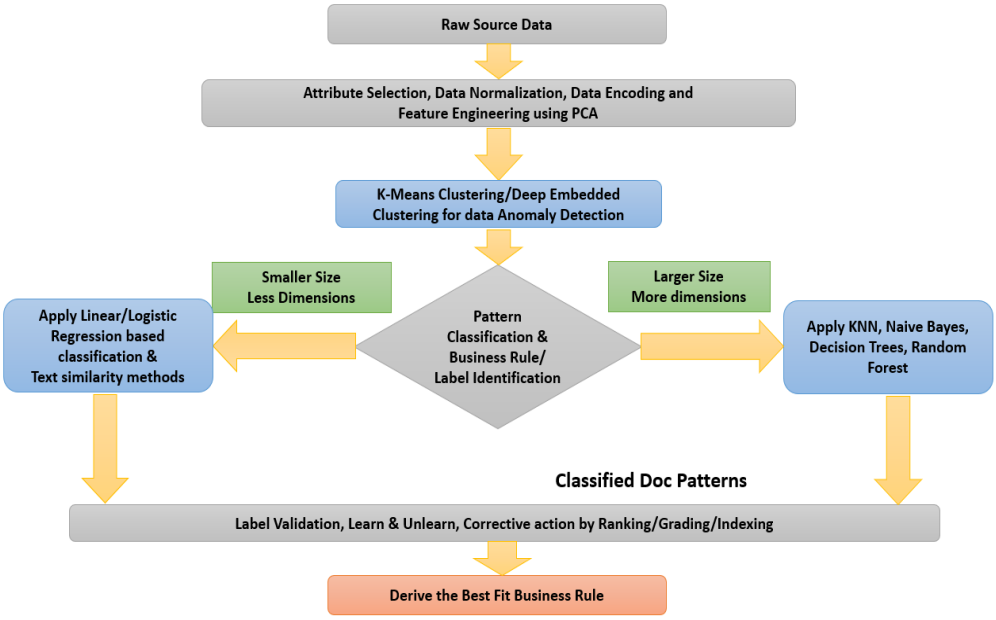


Fig. 3. Hybrid evaluation framework.

Figure 4 represents the NN-based business rule engine which can be used for constructing the NN for unknown data and patterns. Two types of learning, inductive and deductive, are considered when dealing with business rules. Data

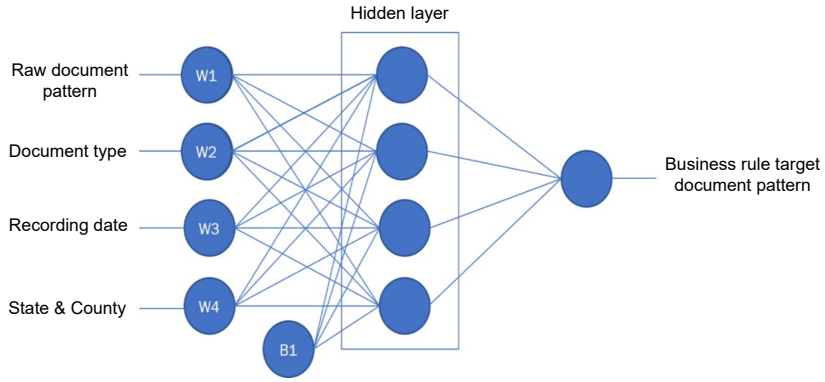


FIG. 4. NN-based business rule engine.

processing in batch and conventional ETL process would only be able to classify known patterns effectively [10]. Any change in the expected pattern or unexpected patterns may enable a default logic that is not always optimal or may require an altogether new business rule logic to be coded to be able to handle such variations. In both cases, it causes latency and availability of the latest data that may impact business greatly, especially companies in the banking sector, insurance domain, etc. Businesses rely more on the latest information than the older data. Thus, the need for real-time data transformation is inevitable to address the challenges of data integration and changing data. With the real-time processing concept, the system should handle the varied formats dynamically. It should be able to map either existing business rule or build a new business rule that can transform the data to the most desirable format. The possibility of having data quality issues in the dynamic system is much higher. So, the need to identify and derive a business rule in real-time for these systems is also greater. Predicting and classifying dynamically the data transformation rules can help in correctly identifying the best-fit business rule and format the new patterns into the business required way without human intervention. In the absence of a dynamic rule engine, the system applies the existing or default rules coded when a new pattern or partially known patterns are received and may not adhere to the business requirements, which may require post-update corrections that are again very time-consuming and require human efforts. Having said that, an evaluation mechanism using various machine learning algorithms should also be incorporated. Based on the complexity of the data, one can decide to apply the right NN along with activation function for business rule classification.

This activation function is used to choose the effective and accurate business rule for complex data with smaller error rate. The depending attributes on which the format of the registered number is contingent, are collected and considered as input to the NN. This includes the region, date in which the registered document

is recorded, type of the registered document, last processed registered document, etc. The registered number format usually depends on any or all of these.

7. Experimental evaluation

For the document type data where the source data format is the registered number varying by the time, a stratified random sampling is best suited to cover all the possible source and target formats. Hence, the data is first grouped by regions that follow the same formats and then further subdivided into strata by month and year and a random sampling mechanism is done to pull at least five documents from each stratum in every region. As seen in Fig. 5, it is necessary to format the source and target pattern in order to qualify the patterns effectively. For pattern format, the registered documents have alphabet letters and numeric values with special characters such as "–", "/", ".", etc. As depicted in the figure, the numeric values are converted to the letter "N", all the letters are converted to "A", and the special characters are forwarded as is. This patterning is applied to both source and target. Since there would be too many patterns to consider only the most frequent patterns, a threshold number is fixed and in this study it is 100. The source and target patterns are grouped by their pattern to obtain a unique pattern per region and only unique patterns that qualify for the minimum threshold value of 100, in this case, are considered. Most of the incorrect formats or very rare formats are eliminated from the final list of patterns.

CCYYSSNNNNNN	1853558
CCYYSSSNNNNN	112240
NNNNNNNN	45288
NNNNNNSSNNN	38568
NNNNNNSNNNN	38072
ANNNNNNNN	24548
CCYYSNNNNNNN	21412
NNNNZNSNNNN	17315
NNNNZNSSNNN	16072
CCYYSSSSNNNN	6681
NNNNZNNN	4341
NNNNNNSSSNN	3337
NNNNZNSSSNN	1601
CCYYSSSSSNNN	1255
NNNNNNN	1222
ANNNZNNNN	1183
ANNNNNNN	932
NNNNZZNN	422
NNNNNNSSSSN	383
ANNNNNN	296
NNNNSSNNNNNN	190
NNNNZNSSSSN	156
CCYYSSSSSSNN	131
NNNNZNN	116

Fig. 5. Formatted patterns.

As part of the experimental evaluation, many classification models were explored and applied to find the optimal clusters by using the qualified region registered document patterns. The elbow curve, silhouette and gap statistics methods were used to identify the optimal number of clusters. The confusion matrix using a decision tree classifier on the training and test data is shown in

Appendix A, which has detailed information on the results. The sampled patterns and confusion matrix is shown in Appendix A. From the first set of results, it is found that the decision tree method provides an accuracy of 93.93% for training data and 94.63% for test data. Appendix B shows the results of the application of random forest with grid search CV to both training and test data.

The random forest with grid search CV provides an improved accuracy of 94.80% and 94.18% for training and test data, respectively. The logistic regression provides 84.38% accuracy for training data and 84.88% for test data, which is shown in Appendix C. Naive Bayes method was not as appropriate as it provided an accuracy of 65.37% for the training data and 66.69% for the test data [11]. The confusion matrix for this evaluation is shown in Appendix D. Apart from the decision tree, random forest, and naive Bayes, this work also considered applying a support vector machine (SVM) to find its effectiveness over the domain data. The results, shown in Appendix E, present the confusion matrix and sampling of data. The SVM method only gave 82.71% accuracy for the training data and 66.69% for the test data. The comparison of accuracy using logistic regression, SVM, decision tree, random forest and naive Bayes by applying different encoding mechanisms is depicted in Table 3.

Table 3. Comparison of accuracy for data transformation using various ML models.

ML model	One hot encoding		Label encoding		Binary encoding	
	Training accuracy [%]	Test accuracy [%]	Training accuracy [%]	Test accuracy [%]	Training accuracy [%]	Test accuracy [%]
Logistic regression	92	91	58	58	84.38	84.88
Support vector machine	92	92	57	57	82.71	66.69
Decision tree	93	92	95	90	93.93	94.65
Random forest	90	89	94	92	94.80	94.18
Naive Bayes	49	49	26	26	65.37	66.69

8. Conclusion and future directions

This paper presented the introduction and all the required research context for the purpose of designing the ensemble methodology for supporting the classification of business rules for title registered documents. The proposed model of building a dynamic business rule engine and the suggested methodology can be applied further and evaluated for registered documents from various regions. Also, in this paper, various classification models were used to properly segment and classify the output data correctly by using the statistical parameters such as precision, accuracy, and F-score. Among various classification models, decision

tree and random forest with grid search CV outperformed logistic regression, naive Bayes and SVM with grid search CV. This work illustrated the benefit of applying the hybrid ensemble technique, and found that for the complex text-based title registered documents random forest and decision tree methods provided optimal performance based on the metrics considered for the evaluation. Application of activation functions and NN-based data engine can be used for both data preprocessing and mapping, which can be further evaluated to find better performance. This work can also be customized for the source and target patterns through some APIs for clustering similar documents to title industry documents. Apart from increasing the volume of input data, varying training and test data distribution percentages can also be considered for validating the consistency of the proposed methodology.

References

1. L. Contreras-Ochando, C. Ferri, J. Hernández-Orallo, F. Martínez-Plumed, M.J. Ramírez-Quintana, S. Katayama, Automated data transformation with inductive programming and dynamic background knowledge, [in:] *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019*, Würzburg, Germany, September 16–20, Proceedings, Part III, pp. 735–751, 2020, doi: 10.1007/978-3-030-46133-1_44.

2. W. Kratsch, J. Manderscheid, M. Röglinger, J. Seyfried, Machine learning in business process monitoring: A comparison of deep learning and classical approaches used for outcome prediction, *Business and Information Systems Engineering*, **63**: 261–276, 2021, doi: 10.1007/s12599-020-00645-0.

3. A. Guha, D. Samanta, Hybrid approach to document anomaly detection: an application to facilitate RPA in title insurance, *International Journal of Automation and Computing*, **18**: 55–72, 2021, doi: 10.1007/s11633-020-1247-y.

4. S. Sun, Z. Cao, H. Zhu, J. Zhao, A survey of optimization methods from a machine learning perspective, *IEEE Transactions on Cybernetics*, **50**(8): 3668–3681, 2020, doi: 10.1109/TCYB.2019.2950779.

5. R. Nagar, Y. Singh, A literature survey on machine learning algorithms, *Journal of Emerging Technologies and Innovative Research*, **6**(4): 471–474, 2019, https://www.jetir.org/view?paper=JETIR1904C77.

6. D. Ndirangu, W. Mwangi, L. Nderu, A hybrid ensemble method for multiclass classification and outlier detection, *International Journal of Sciences: Basic and Applied Research*, **45**(1): 192–213, 2019, https://www.gssrr.org/index.php/JournalOfBasicAndApplied/article/view/9904.

7. N.A. Wahid, T.N. Adi, H. Bae, Y. Choi, Predictive business process monitoring – Remaining time prediction using deep neural network with entity embedding, [in:] *The Fifth Information Systems International Conference: Elsevier Procedia Computer Science*, Surabaya, Indonesia, July 23–24, Vol. 161, pp. 1080–1088, 2019, doi: 10.1016/j.procs.2019.11.219.

8. R. Makani, B.V.R. Reddy, Taxonomy of machine leaning based anomaly detection and its suitability, [in:] *International Conference on Computational Intelligence and Data Science:*

Elsevier Procedia Computer Science, Ohio, USA, April 7–8, Vol. 132, pp. 1842–1849, 2018, doi: 10.1016/j.procs.2018.05.133.

9. J. Zhang, Advancements of outlier detection: A survey, *ICST Transactions on Scalable Information Systems*, **13**(1–3): 1–26, 2013, doi: 10.4108/trans.sis.2013.01-03.e2.

10. N. Biswas, S. Chattapadhyay, G. Mahapatra, S. Chatterjee, K.C. Mondal, A new approach for conceptual ETL process modeling, *International Journal of Ambient Computing and Intelligence*, **10**(1): 30–45, 2009, doi: 10.4018/IJACI.2019010102.

11. G.M.D. Sree, S. Vasundra, Vector-based classification prediction to geographical location, *International Journal of Future Generation Communication and Networking*, **13**(4): 4174–4179, 2020.

Appendix A. Confusion matrix using decision tree classifier on training and test data

Accuracy: 93.93
Confusion Matrix:

```
[[  459     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0]
 [    0  1006     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0]
 [    0     0  1229     0     0     0     0     0     2     0     0     0     0
     0     0     0     0     0]
 [    0     0     0 17991    21     3     0     3     3     0     0     0     0
     0     2     0     0     0]
 [    0     0     0    83  4399     0     0     0     2     0     0     0     0
     0     1     0     0     0]
 [    0     0     0     0     2   419     0     1     3     0     0     0     0
     0     0     0     0     0]
 [    0     0     0     0     0     0   510     0     1     0     0     0     0
     0     0     0     0     0]
 [    0     0     0     1     0     0     2  2145     3     0     0     0     0
     0     0     0     0     0]
 [    0     0     0     1     0     0     0     0   802     0     0     0     0
     0     0     0     0     0]
 [    0     0     0     1     0     0     0     0   712     7     0     0     0
     0     0     0     0     0]
 [    0     0     0     1     0     0     0     0   240     0     0     0     0
     0     0     0     0     0]
 [    0     0     0     0     0     0     0     0   104     0     0     0     0
     0     0     0     0     0]
 [    0     0     0     0     0     0     0   266     0     0     0     0     0
    49     0     0     0     0]
 [    0     0     0     0     0     0     0     0     1     1     0     0     0
     0   310     0     0     0]
 [    0     0     0     0     0     0     0     0     2     1     0     0     0
     0   257     0     0     0]
 [    0     0     0     0     0     0     0     0     0     0     0     0     0
     0   105     0     0     0]
 [    0     0     0     0     0     0     0    71     0     0     0     0     0
     0     0     0     0     2]]
```

	precision	recall	f1-score	support
ANNNNNNN	1.00	1.00	1.00	459
ANNNNNNNN	1.00	1.00	1.00	1006
CCYYSNNNNNN	1.00	1.00	1.00	1231
CCYYSSNNNNN	1.00	1.00	1.00	18023
CCYYSSSNNNN	0.99	0.98	0.99	4485
CCYYSSSSNNN	0.99	0.99	0.99	425
NNNNNNN	1.00	1.00	1.00	511
NNNNNNNN	0.86	1.00	0.93	2151
NNNNNNSNNN	0.43	1.00	0.60	803
NNNNNNSSNN	0.78	0.01	0.02	720
NNNNNNSSSNN	0.00	0.00	0.00	241
NNNNNNSSSSN	0.00	0.00	0.00	104
NNNNZNNN	1.00	0.16	0.27	315
NNNNZNSNNN	0.46	0.99	0.63	312
NNNNZNSSNN	0.00	0.00	0.00	260
NNNNZNSSSNN	0.00	0.00	0.00	105
NNNNZZNN	1.00	0.03	0.05	73
accuracy			0.94	31224
macro avg	0.68	0.60	0.56	31224
weighted avg	0.94	0.94	0.92	31224

Accuracy: 94.65
Confusion Matrix:

```
[[ 117    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   0  229    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   0    0  276    0    0    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   0    0    0 4574    6    2    0    0    0    1    0    0    0    1
     0    0    0]
 [   0    0    0   22 1121    2    0    0    0    0    0    0    0    2
     0    0    0]
 [   0    0    0    0    1  105    0    0    0    0    0    0    0    0
     0    0    0]
 [   0    0    0    0    0    0  133    0    1    0    0    0    0    0
     0    0    0]
 [   0    0    0    1    0    0    0  524    1    0    0    0    0    0
     0    0    0]
 [   0    0    0    0    0    0    0    1  206    0    0    0    0    0
     0    0    0]
 [   0    0    0    0    0    0    0    0    0  161    2    0    0    0
     0    0    0]
 [   0    0    0    0    0    0    0    0    0   46    0    0    0    0
     0    0    0]
 [   0    0    0    0    0    0    0    0    0   25    0    0    0    0
     0    0    0]
 [   0    0    0    0    0    0    0    0    0   48    0    0    0    0
    14    0    0]
 [   0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0   86]
 [   0    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0   60]
 [   0    0    0    0    0    0    0    0    0    1    0    0    0    0
     0    0   24]
 [   0    0    0    0    0    0    0    0    0   12    0    0    0    0
     0    0    0]
 [   0    0    1]]
```

	precision	recall	f1-score	support
ANNNNNNN	1.00	1.00	1.00	117
ANNNNNNNN	1.00	1.00	1.00	229
CCYYSNNNNNN	1.00	1.00	1.00	276
CCYYSSNNNNN	0.99	1.00	1.00	4584
CCYYSSSNNNN	0.99	0.98	0.99	1147
CCYYSSSSNNN	0.96	0.99	0.98	106
NNNNNNN	1.00	0.99	1.00	134
NNNNNNNN	0.90	1.00	0.94	526
NNNNNNSNNN	0.47	1.00	0.64	207
NNNNNNSSNN	0.67	0.01	0.02	163
NNNNNNSSSNN	0.00	0.00	0.00	46
NNNNNNSSSSN	0.00	0.00	0.00	25
NNNNZNNN	1.00	0.23	0.37	62
NNNNZNSNNN	0.50	1.00	0.66	86
NNNNZNSSNN	0.00	0.00	0.00	60
NNNNZNSSSNN	0.00	0.00	0.00	25
NNNNZZNN	1.00	0.08	0.14	13
accuracy			0.95	7806
macro avg	0.68	0.60	0.57	7806
weighted avg	0.94	0.95	0.93	7806

Appendix B. Confusion matrix using random forest with grid search CV

Accuracy: 94.80
Confusion Matrix:

```
[[ 458    1    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0]
 [   0 1006    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0]
 [   0    0 1229    2    0    0    0    0    0    0    0    0
     0    0    0    0    0]
 [   0    0    0 18022    1    0    0    0    0    0    0    0
     0    0    0    0    0]
 [   0    0    0   28 4457    0    0    0    0    0    0    0
     0    0    0    0    0]
 [   0    0    0    0    0  425    0    0    0    0    0    0
     0    0    0    0    0]
 [   0    0    0    0    0    0  509    2    0    0    0    0
     0    0    0    0    0]
 [   0    0    0    0    0    0    0 2151    0    0    0    0
     0    0    0    0    0]
 [   0    0    0    5    1    0    0    0  653  144    0    0
     0    0    0    0    0]
 [   0    0    0    3    0    0    0    1  439  277    0    0
     0    0    0    0    0]
 [   0    0    0    0    0    0    0    1  151   89    0    0
     0    0    0    0    0]
 [   0    0    0    0    0    0    0    0   74   30    0    0
     0    0    0    0    0]
 [   0    0    0    0    0    0    0  265    0    0    0    0
    50    0    0    0    0]
 [   0    0    0    2    0    0    0    1    1    0    0    0
     0  264   44    0    0]
 [   0    0    0    3    0    0    0    0    0    1    0    0
     0  157   99    0    0]
 [   0    0    0    1    0    0    0    0    0    0    0    0
     0   79   25    0    0]
 [   0    0    0    0    0    0    0   73    0    0    0    0
     0    0    0    0    0]]
```

	precision	recall	f1-score	support
ANNNNNN	1.00	1.00	1.00	459
ANNNNNNN	1.00	1.00	1.00	1006
CCYYSNNNNNN	1.00	1.00	1.00	1231
CCYYSSNNNNN	1.00	1.00	1.00	18023
CCYYSSSNNNN	1.00	0.99	1.00	4485
CCYYSSSSNNN	1.00	1.00	1.00	425
NNNNNN	1.00	1.00	1.00	511
NNNNNNN	0.86	1.00	0.93	2151
NNNNNSNNN	0.50	0.81	0.62	803
NNNNNSSNN	0.51	0.38	0.44	720
NNNNNSSSN	0.00	0.00	0.00	241
NNNNNSSSSN	0.00	0.00	0.00	104
NNNNZNNN	1.00	0.16	0.27	315
NNNNZNSNNN	0.53	0.85	0.65	312
NNNNZNSSNNN	0.59	0.38	0.46	260
NNNNZNSSSNN	0.00	0.00	0.00	105
NNNNZZNN	0.00	0.00	0.00	73
accuracy			0.95	31224
macro avg	0.65	0.62	0.61	31224
weighted avg	0.94	0.95	0.94	31224

Accuracy: 94.18
Confusion Matrix:

```
[[ 117    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   0  229    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   0    0  276    0    0    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   0    0    0 4583    1    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   0    0    0   11 1135    1    0    0    0    0    0    0    0    0
     0    0    0]
 [   0    0    0    0    0  106    0    0    0    0    0    0    0    0
     0    0    0]
 [   0    0    0    0    0    0  133    1    0    0    0    0    0    0
     0    0    0]
 [   0    0    0    0    0    0    0  526    0    0    0    0    0    0
     0    0    0]
 [   0    0    0    2    0    0    0    1  128   76    0    0    0    0
     0    0    0]
 [   0    0    0    1    0    0    0    0  120   42    0    0    0    0
     0    0    0]
 [   0    0    0    0    0    0    0    0   26   20    0    0    0    0
     0    0    0]
 [   0    0    0    0    0    0    0    0   10   15    0    0    0    0
     0    0    0]
 [   0    0    0    0    0    0    0   50    0    0    0    0   12    0
     0    0    0]
 [   0    0    0    3    0    0    0    0    0    0    0    0    0   55
    28    0    0]
 [   0    0    0    0    0    0    0    0    0    0    0    0    0   50
    10    0    0]
 [   0    0    0    1    0    0    0    0    0    0    0    0    0   21
     3    0    0]
 [   0    0    0    0    0    0    0   13    0    0    0    0    0    0
     0    0    0]]
```

	precision	recall	f1-score	support
ANNNNNN	1.00	1.00	1.00	117
ANNNNNNN	1.00	1.00	1.00	229
CCYYSNNNNNN	1.00	1.00	1.00	276
CCYYSSNNNNN	1.00	1.00	1.00	4584
CCYYSSSNNNN	1.00	0.99	0.99	1147
CCYYSSSSNNN	0.99	1.00	1.00	106
NNNNNN	1.00	0.99	1.00	134
NNNNNNN	0.89	1.00	0.94	526
NNNNNSNNN	0.45	0.62	0.52	207
NNNNNSSNN	0.27	0.26	0.27	163
NNNNNSSSN	0.00	0.00	0.00	46
NNNNNSSSSN	0.00	0.00	0.00	25
NNNNZNNN	1.00	0.19	0.32	62
NNNNZNSNNN	0.44	0.64	0.52	86
NNNNZNSSNNN	0.24	0.17	0.20	60
NNNNZNSSSNN	0.00	0.00	0.00	25
NNNNZZNN	0.00	0.00	0.00	13
accuracy			0.94	7806
macro avg	0.60	0.58	0.57	7806
weighted avg	0.93	0.94	0.94	7806

Appendix C. Confusion matrix using logistic regression with grid search CV

Accuracy: 84.38
Confusion Matrix:

```
[[ 448    4    0    2    5    0    0    0    0    0    0    0
     0    0    0    0    0]
 [   2  949    0   11   43    0    1    0    0    0    0    0
     0    0    0    0    0]
 [   0    0 1044  187    0    0    0    0    0    0    0    0
     0    0    0    0    0]
 [   0    2   16 17101  889    2    1   10    1    0    0    0
     0    0    1    0    0]
 [   0    0    0 1142 3327   13    0    3    0    0    0    0
     0    0    0    0    0]
 [   0    0    0   54   74  296    0    1    0    0    0    0
     0    0    0    0    0]
 [   0    0    0   14    1    0  409   87    0    0    0    0
     0    0    0    0    0]
 [   0    0    3  182   12    0   88 1866    0    0    0    0
     0    0    0    0    0]
 [   0    0    0   40   77    4    0    1  495  186    0    0
     0    0    0    0    0]
 [   0    0    0   36   74    0    0    1  403  206    0    0
     0    0    0    0    0]
 [   0    0    0   15   32    0    0    0  114   80    0    0
     0    0    0    0    0]
 [   0    0    0    6   15    0    0    0   51   32    0    0
     0    0    0    0    0]
 [   0    0    7   47    2    0    7  252    0    0    0    0
     0    0    0    0    0]
 [   0    0    0   15   85    1    0    1   13    0    0    0
     0  170   27    0    0]
 [   0    0    0   13   49    0    0    1   11    0    0    0
     0  151   35    0    0]
 [   0    0    0    4   31    0    0    0    0    0    0    0
     0   55   15    0    0]
 [   0    0    1    1    0    0    1   70    0    0    0    0
     0    0    0    0    0]]
```

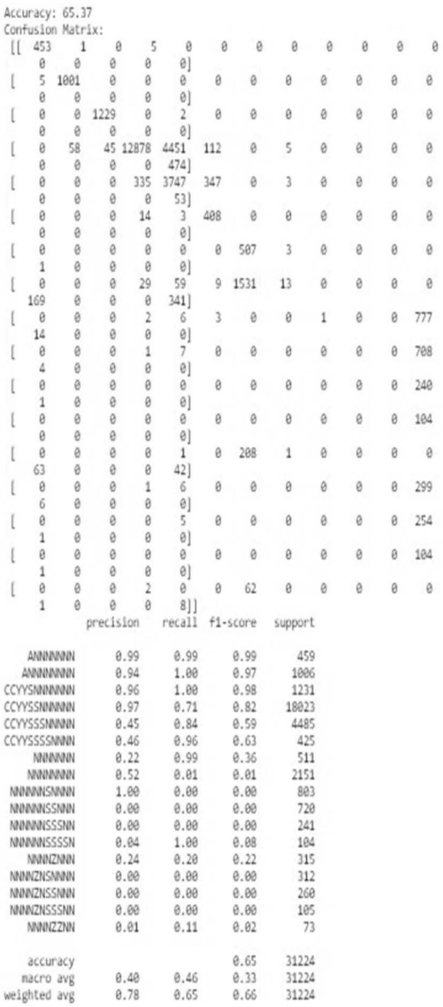
	precision	recall	f1-score	support
ANNNNNNN	1.00	0.98	0.99	459
ANNNNNNNN	0.99	0.94	0.97	1006
CCYYSNNNNNN	0.97	0.85	0.91	1231
CCYYSSNNNNN	0.91	0.95	0.93	18023
CCYYSSSNNNN	0.71	0.74	0.72	4485
CCYYSSSSNNN	0.94	0.70	0.80	425
NNNNNN	0.81	0.80	0.80	511
NNNNNNN	0.81	0.87	0.84	2151
NNNNNSNNN	0.45	0.62	0.52	803
NNNNNSSNN	0.41	0.29	0.34	720
NNNNNSSSN	0.00	0.00	0.00	241
NNNNNSSSSN	0.00	0.00	0.00	104
NNNZNNN	0.00	0.00	0.00	315
NNNZNSNNN	0.45	0.54	0.49	312
NNNZNSSNN	0.45	0.13	0.21	260
NNNZNSSSN	0.00	0.00	0.00	105
NNNZZNN	0.00	0.00	0.00	73
accuracy			0.84	31224
macro avg	0.52	0.49	0.50	31224
weighted avg	0.82	0.84	0.83	31224

Accuracy: 84.88
Confusion Matrix:

```
[[ 113    1    0    1    2    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   2  212    0    2   11    0    2    0    0    0    0    0    0    0
     0    0    0]
 [   0    0  238   38    0    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   0    0    3 4375  205    0    0    1    0    0    0    0    0    0
     0    0    0]
 [   0    0    0  286  860    1    0    0    0    0    0    0    0    0
     0    0    0]
 [   0    0    0    8   21   76    0    1    0    0    0    0    0    0
     0    0    0]
 [   0    0    0    7    0    0  103   24    0    0    0    0    0    0
     0    0    0]
 [   0    0    1   52    4    0   31  438    0    0    0    0    0    0
     0    0    0]
 [   0    0    0    8   27    0    0    0  121   51    0    0    0    0
     0    0    0]
 [   0    0    0    9   18    0    0    0   99   37    0    0    0    0
     0    0    0]
 [   0    0    0    1    7    0    0    0   24   14    0    0    0    0
     0    0    0]
 [   0    0    0    3    7    0    0    0    6    9    0    0    0    0
     0    0    0]
 [   0    0    1   14    0    0    3   44    0    0    0    0    0    0
     0    0    0]
 [   0    0    0   10   15    0    0    0    7    0    0    0    0   46
     8    0    0]
 [   0    0    0    3   11    0    0    0    3    0    0    0    0   36
     7    0    0]
 [   0    0    0    3    6    0    0    0    0    0    0    0    0   15
     1    0    0]
 [   0    0    0    1    0    0    0   12    0    0    0    0    0    0
     0    0    0]]
```

	precision	recall	f1-score	support
ANNNNNNN	0.98	0.97	0.97	117
ANNNNNNNN	1.00	0.93	0.96	229
CCYYSNNNNNN	0.98	0.86	0.92	276
CCYYSSNNNNN	0.91	0.95	0.93	4584
CCYYSSSNNNN	0.72	0.75	0.73	1147
CCYYSSSSNNN	0.99	0.72	0.83	106
NNNNNN	0.74	0.77	0.75	134
NNNNNNN	0.84	0.83	0.84	526
NNNNNSNNN	0.47	0.58	0.52	207
NNNNNSSNN	0.33	0.23	0.27	163
NNNNNSSSN	0.00	0.00	0.00	46
NNNNNSSSSN	0.00	0.00	0.00	25
NNNZNNN	0.00	0.00	0.00	62
NNNZNSNNN	0.47	0.53	0.50	86
NNNZNSSNN	0.44	0.12	0.18	60
NNNZNSSSN	0.00	0.00	0.00	25
NNNZZNN	0.00	0.00	0.00	13
accuracy			0.85	7806
macro avg	0.52	0.48	0.49	7806
weighted avg	0.83	0.85	0.84	7806

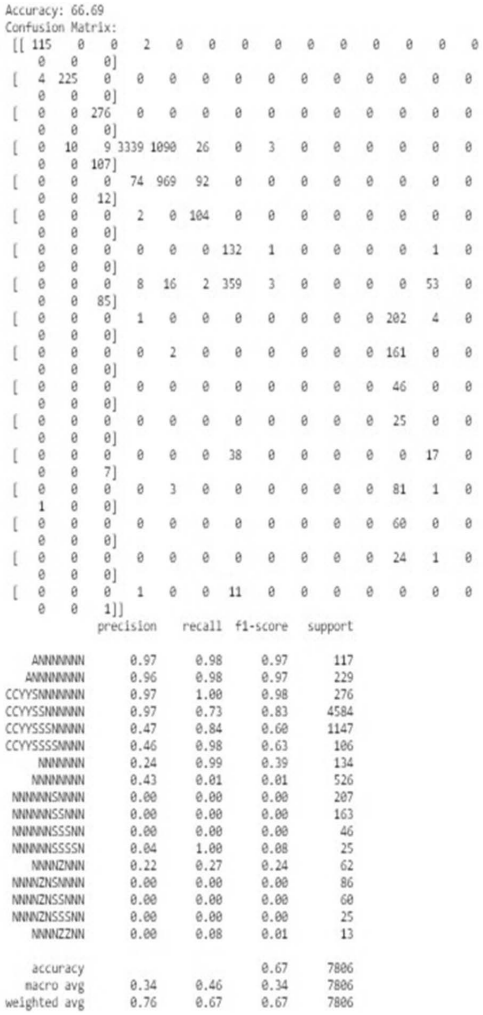
Appendix D. Confusion matrix using naive Bayes on training and test data



```
Accuracy: 65.37
Confusion Matrix:
[[  453     1     0     5     0     0     0     0     0     0     0     0
      0     0     0     0     0]
 [    5  1001     0     0     0     0     0     0     0     0     0     0
      0     0     0     0     0]
 [    0     0  1229     0     2     0     0     0     0     0     0     0
      0     0     0     0     0]
 [    0    58    45 12878  4451   112     0     5     0     0     0     0
      0     0     0     0   474]
 [    0     0     0   335  3747   347     0     3     0     0     0     0
      0     0     0     0    53]
 [    0     0     0    14     3   408     0     0     0     0     0     0
      0     0     0     0     0]
 [    0     0     0     0     0     0   507     3     0     0     0     0
      1     0     0     0     0]
 [    0     0     0    29    59     9  1531    13     0     0     0     0
    169     0     0     0   341]
 [    0     0     0     2     6     3     0     0     1     0     0   777
     14     0     0     0     0]
 [    0     0     0     1     7     0     0     0     0     0     0   708
      4     0     0     0     0]
 [    0     0     0     0     0     0     0     0     0     0     0   240
      1     0     0     0     0]
 [    0     0     0     0     0     0     0     0     0     0     0   104
      0     0     0     0     0]
 [    0     0     0     0     1     0   208     1     0     0     0     0
     63     0     0     0    42]
 [    0     0     0     1     6     0     0     0     0     0     0   299
      6     0     0     0     0]
 [    0     0     0     0     5     0     0     0     0     0     0   254
      1     0     0     0     0]
 [    0     0     0     0     0     0     0     0     0     0     0   104
      1     0     0     0     0]
 [    0     0     0     2     0     0    62     0     0     0     0     0
      1     0     0     0     8]]
              precision    recall  f1-score   support

    ANNNNNNN       0.99      0.99      0.99       459
   ANNNNNNNN       0.94      1.00      0.97      1006
CCYYSNNNNNNN       0.96      1.00      0.98      1231
CCYYSSNNNNNN       0.97      0.71      0.82     18023
CCYYSSSNNNNN       0.45      0.84      0.59      4485
CCYYSSSSNNNN       0.46      0.96      0.63       425
      NNNNNN       0.22      0.99      0.36       511
     NNNNNNN       0.52      0.01      0.01      2151
  NNNNNNSNNN       1.00      0.00      0.00       803
  NNNNNNSSNN       0.00      0.00      0.00       720
  NNNNNNSSSN       0.00      0.00      0.00       241
  NNNNNNSSSSN      0.04      1.00      0.08       104
     NNNNZNNN      0.24      0.20      0.22       315
  NNNNZNSNNN       0.00      0.00      0.00       312
  NNNNZNSSNN       0.00      0.00      0.00       260
  NNNNZNSSSN       0.00      0.00      0.00       105
     NNNNZZNN      0.01      0.11      0.02        73

     accuracy                           0.65     31224
    macro avg      0.40      0.46      0.33     31224
 weighted avg      0.78      0.65      0.66     31224
```

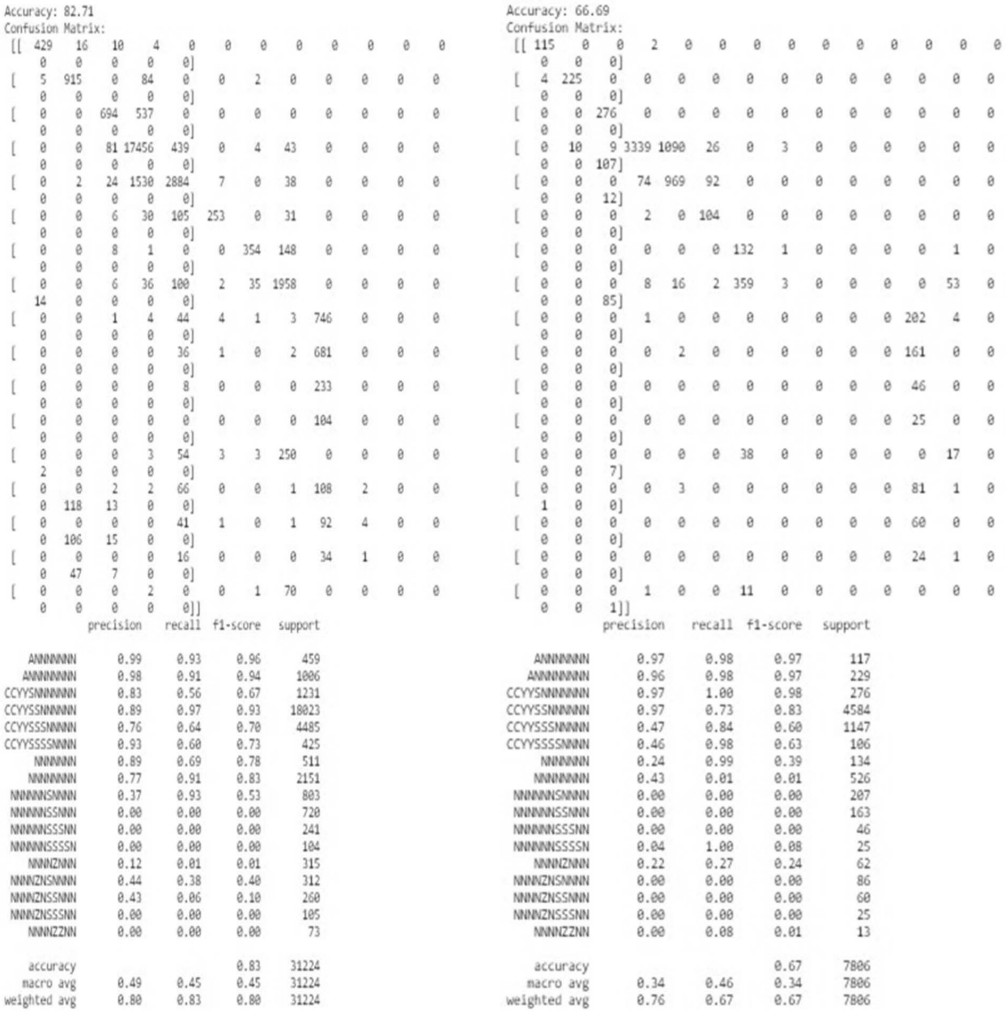


```
Accuracy: 66.69
Confusion Matrix:
[[ 115    0    0    2    0    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   4  225    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   0    0  276    0    0    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   0   10    9 3339 1090   26    0    3    0    0    0    0    0    0
     0    0  107]
 [   0    0    0   74  969   92    0    0    0    0    0    0    0    0
     0    0   12]
 [   0    0    0    2    0  104    0    0    0    0    0    0    0    0
     0    0    0]
 [   0    0    0    0    0    0  132    1    0    0    0    0    1    0
     0    0    0]
 [   0    0    0    8   16    2  359    3    0    0    0    0   53    0
     0    0   85]
 [   0    0    0    1    0    0    0    0    0    0    0    0  202    4    0
     0    0    0]
 [   0    0    0    0    2    0    0    0    0    0    0    0  161    0    0
     0    0    0]
 [   0    0    0    0    0    0    0    0    0    0    0    0   46    0    0
     0    0    0]
 [   0    0    0    0    0    0    0    0    0    0    0    0   25    0    0
     0    0    0]
 [   0    0    0    0    0    0   38    0    0    0    0    0    0   17    0
     0    0    7]
 [   0    0    0    0    3    0    0    0    0    0    0    0   81    1    0
     1    0    0]
 [   0    0    0    0    0    0    0    0    0    0    0    0   60    0    0
     0    0    0]
 [   0    0    0    0    0    0    0    0    0    0    0    0   24    1    0
     0    0    0]
 [   0    0    0    1    0    0   11    0    0    0    0    0    0    0    0
     0    0    1]]
              precision    recall  f1-score   support

    ANNNNNNN       0.97      0.98      0.97       117
   ANNNNNNNN       0.96      0.98      0.97       229
CCYYSNNNNNNN       0.97      1.00      0.98       276
CCYYSSNNNNNN       0.97      0.73      0.83      4584
CCYYSSSNNNNN       0.47      0.84      0.60      1147
CCYYSSSSNNNN       0.46      0.98      0.63       106
      NNNNNN       0.24      0.99      0.39       134
     NNNNNNN       0.43      0.01      0.01       526
  NNNNNNSNNN       0.00      0.00      0.00       207
  NNNNNNSSNN       0.00      0.00      0.00       163
  NNNNNNSSSN       0.00      0.00      0.00        46
  NNNNNNSSSSN      0.04      1.00      0.08        25
     NNNNZNNN      0.22      0.27      0.24        62
  NNNNZNSNNN       0.00      0.00      0.00        86
  NNNNZNSSNN       0.00      0.00      0.00        60
  NNNNZNSSSN       0.00      0.00      0.00        25
     NNNNZZNN      0.00      0.08      0.01        13

     accuracy                           0.67      7806
    macro avg      0.34      0.46      0.34      7806
 weighted avg      0.76      0.67      0.67      7806
```

Appendix E. Confusion matrix using SVM with grid search CV on training and test data



```
Accuracy: 82.71
Confusion Matrix:
[[ 429   16   10    4    0    0    0    0    0    0    0    0
     0    0    0    0    0]
 [   5  915    0   84    0    0    2    0    0    0    0    0
     0    0    0    0    0]
 [   0    0  694  537    0    0    0    0    0    0    0    0
     0    0    0    0    0]
 [   0    0   81 17456  439    0    4   43    0    0    0    0
     0    0    0    0    0]
 [   0    2   24 1530 2884    7    0   38    0    0    0    0
     0    0    0    0    0]
 [   0    0    6   30  105  253    0   31    0    0    0    0
     0    0    0    0    0]
 [   0    0    8    1    0    0  354  148    0    0    0    0
     0    0    0    0    0]
 [   0    0    6   36  100    2   35 1958    0    0    0    0
    14    0    0    0    0]
 [   0    0    1    4   44    4    1    3  746    0    0    0
     0    0    0    0    0]
 [   0    0    0    0   36    1    0    2  681    0    0    0
     0    0    0    0    0]
 [   0    0    0    0    8    0    0    0  233    0    0    0
     0    0    0    0    0]
 [   0    0    0    0    0    0    0    0  104    0    0    0
     0    0    0    0    0]
 [   0    0    0    3   54    3    3  250    0    0    0    0
     2    0    0    0    0]
 [   0    0    2    2   66    0    0    1  108    2    0    0
     0  118   13    0    0]
 [   0    0    0    0   41    1    0    1   92    4    0    0
     0  106   15    0    0]
 [   0    0    0    0   16    0    0    0   34    1    0    0
     0   47    7    0    0]
 [   0    0    0    2    0    0    1   70    0    0    0    0
     0    0    0    0    0]]
              precision    recall  f1-score   support

    ANNNNNN       0.99      0.93      0.96       459
   ANNNNNNN       0.98      0.91      0.94      1006
CCYYSNNNNNN       0.83      0.56      0.67      1231
CCYYSSNNNNN       0.89      0.97      0.93     18023
CCYYSSSNNNN       0.76      0.64      0.70      4485
CCYYSSSSNNN       0.93      0.60      0.73       425
     NNNNNN       0.89      0.69      0.78       511
    NNNNNNN       0.77      0.91      0.83      2151
NNNNNNSNNN        0.37      0.93      0.53       803
NNNNNNSSNN        0.00      0.00      0.00       720
NNNNNNSSSN        0.00      0.00      0.00       241
NNNNNNSSSSN       0.00      0.00      0.00       104
   NNNNZNNN       0.12      0.01      0.01       315
NNNNZNSNNN        0.44      0.38      0.40       312
NNNNZNSSNN        0.43      0.06      0.10       260
NNNNZNSSSN        0.00      0.00      0.00       105
   NNNNZZNN       0.00      0.00      0.00        73

    accuracy                           0.83     31224
   macro avg       0.49      0.45      0.45     31224
weighted avg       0.80      0.83      0.80     31224
```

```
Accuracy: 66.69
Confusion Matrix:
[[ 115    0    0    2    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   4  225    0    0    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   0    0  276    0    0    0    0    0    0    0    0    0    0
     0    0    0]
 [   0   10    9 3339 1090   26    0    3    0    0    0    0    0
     0    0  107]
 [   0    0    0   74  969   92    0    0    0    0    0    0    0
     0    0   12]
 [   0    0    0    2    0  104    0    0    0    0    0    0    0
     0    0    0]
 [   0    0    0    0    0    0  132    1    0    0    0    0    1
     0    0    0]
 [   0    0    0    8   16    2  359    3    0    0    0    0   53
     0    0   85]
 [   0    0    0    1    0    0    0    0    0    0    0  202    4
     0    0    0]
 [   0    0    0    0    2    0    0    0    0    0    0  161    0
     0    0    0]
 [   0    0    0    0    0    0    0    0    0    0    0   46    0
     0    0    0]
 [   0    0    0    0    0    0    0    0    0    0    0   25    0
     0    0    0]
 [   0    0    0    0    0    0   38    0    0    0    0    0   17
     0    0    7]
 [   0    0    0    0    3    0    0    0    0    0    0   81    1
     1    0    0]
 [   0    0    0    0    0    0    0    0    0    0    0   60    0
     0    0    0]
 [   0    0    0    0    0    0    0    0    0    0    0   24    1
     0    0    0]
 [   0    0    0    1    0    0   11    0    0    0    0    0    0
     0    0    1]]
              precision    recall  f1-score   support

    ANNNNNN       0.97      0.98      0.97       117
   ANNNNNNN       0.96      0.98      0.97       229
CCYYSNNNNNN       0.97      1.00      0.98       276
CCYYSSNNNNN       0.97      0.73      0.83      4584
CCYYSSSNNNN       0.47      0.84      0.60      1147
CCYYSSSSNNN       0.46      0.98      0.63       106
     NNNNNN       0.24      0.99      0.39       134
    NNNNNNN       0.43      0.01      0.01       526
NNNNNNSNNN        0.00      0.00      0.00       207
NNNNNNSSNN        0.00      0.00      0.00       163
NNNNNNSSSN        0.00      0.00      0.00        46
NNNNNNSSSSN       0.04      1.00      0.08        25
   NNNNZNNN       0.22      0.27      0.24        62
NNNNZNSNNN        0.00      0.00      0.00        86
NNNNZNSSNN        0.00      0.00      0.00        60
NNNNZNSSSN        0.00      0.00      0.00        25
   NNNNZZNN       0.00      0.08      0.01        13

    accuracy                           0.67      7806
   macro avg       0.34      0.46      0.34      7806
weighted avg       0.76      0.67      0.67      7806
```

Received January 15, 2022; revised version March 30, 2022; accepted May 18, 2022.