Comparative Analysis of Machine Learning Methods Application for Financial Fraud Detection

Alexander Menshchikov, Vladislav Perfilev, Denis Roenko, Maksim Zykin, Maksim Fedosenko
ITMO University
menshikov@itmo.ru {vladislavperfilev1, roenko.dv, makxzm, fedosenkomaksim98}@gmail.com

This paper addresses the fraud detection problem in the context of Big Data used in remote banking systems. The paper aims to propose a new algorithm for automatic detection of fraudulent transactions using machine learning with a performance that allows to apply it in big data systems. The article identifies promising directions for optimizing the operation of methods for fraudulent transactions detection in anti-fraud systems. Architectural approaches to the operation of anti-fraud systems have been studied. Based on this, an architecture for illegal actions prediction in a near real-time mode was proposed. The research task of the article is to find the most suitable machine learning algorithm, with the least training and prediction time, demonstrating high classification performance. To achieve this goal, an analysis of the supervised and ensemble machine learning algorithms was made. The dataset was preprocessed for the experiment with SMOTE resampling and robust scaling techniques. The chosen methods were compared using different metrics: *F*1 score, AUC and time consumption for training and classification. As a result of a metrics comparison, it was found that multilayer perceptron (MLP) and boosting methods (Adaptive, Gradient, XGBoost) has the highest classification, but MLP outperforms boosting methods in terms of time consumption for classification. Thus, MLP was selected as the most appropriate algorithm for further integration to proposed Big Data architecture. Based on the data obtained during the experiments, the degree of their implementation in fraud detection systems was assessed and architecture for the anti-fraud detection system for big data was proposed.

I. Introduction

The beginning of the 21st century has seen a great leap forward in the increasing use of digital information due to ongoing technological advances. Moreover, there has been a rapid growth in the collection of data for the purpose of its analysis and practical application. This trend leads to the term Big Data (BD), which has become widely used since Nature editor Clifford Lynch published a paper in 2008 on current issues in information technology [1]. Big Data is a large volume of digital data with a high update rate, weak structuring, non-relational nature of the database used for processing, with heap storage technology predominating. At the same time dataset can be collected from several sources and have cardinal differences between the data fields included in it. The collected data must be correctly processed and the results correctly interpreted. This task gives rise to such a representative of modern IT science as Data Science.

The modern growth in computing power of computers, together with the development and technical sophistication of Machine Learning (ML) technologies, provides new opportunities for the development and application of artificial intelligence (AI). Currently, machine learning algorithms are widely used in many fields of activity: science, medicine, sports, Internet of things, analytics. However, the progress in this area is almost directly related to improvements in hardware performance, which is now increasingly affected by the physical limitations of computers over time. There is a lot of research and practical work to increase computing resources, optimize their use, distribute the system and work with Big Data. Blockchain technology, Cloud technology, quantum computers are among available solutions for distributed systems. These solutions have a place in Machine Learning, but their predominant role is in the implementation of algorithms, while having no influence on the algorithm structure itself. Therefore, the task of developing the most optimal machine learning algorithms is most relevant when dealing with Big Data. Thus, there is a need to optimize machine learning algorithms and then identify the most accurate, non-resource-intensive, and task-adaptive approaches.

These technologies also have applications in the final- coal sector for the following tasks:

- Identification of potential non-payers and service debtors;
- Credit and consumer scoring;
- Financial crime detection (e.g., sponsorship of terrorism);
- Bank anti-fraud systems and fraud-monitoring [2].

Modern banking anti-fraud systems are steadily introducing integration with machine learning into their framework, in order to deal with the challenges outlined above. The implementation of fraud monitoring is a system designed to evaluate financial transactions for suspicion of fraud. The notion of fraud is often associated with transactions involving the theft of money from bank cards and accounts. However, there are other types of fraud that are aimed at deception and unlawful actions.

As a result, the optimal use of machine learning techniques in anti-fraud systems is a predominant challenge to the financial security of users' wallets and banking organizations in general. It is worth analyzing actual machine learning methods and the possibility to integrate them into bank anti-fraud systems to qualitatively detect fraudulent transactions among a general sample of payments. The research objective is to find promising

directions for optimizing the available approaches and adapting them to real-world conditions, with the possibility of further implementation in existing anti-fraud systems [3].

Then in the paper, in the "Existing approaches" section is presented a literary study of scientific works on the topic of fraud detection using machine learning methods. The "Fraudulent transactions detection" describes the integration of artificial intelligence into service systems, taking into account the indicators of response speed, scalability and fault tolerance. The "Data" section has a description of the dataset processing used in the experiment. The machine learning methods used in the experimental learning are presented in the "Methods" and the "Results" section consist of the final data indicators of experiment and their interpretation for the artificial intelligence problems.

II. Main part

A. Existing approaches

Currently, there are several main approaches to solving the problem of fraudulent transaction detection. The most common tactic is to use different types of neural networks. For example, authors of the article "Algorithms for data mining of banking transactions as part of a system for combating financial fraud" [4] compare multilayer perceptron, random forest and support vector machine classifiers calculating sensitivity, specificity, precision, recall and Matthew's correlation coefficient (MCC). It was found that the classifier based on a random forest showed the best results in terms of a set of the criterial. The applications of algorithms in the works [5],[6],[7],[8], [9] are considered similarly, with the solution of a number of problems, such as data redundancy, inconsistency, noise, heterogeneity and others. In the article [10], the reliability of the model is achieved by combining three sub-methods: Recursive Feature Elimination (RFE), GridSearchCV for Hyperparameter Optimization (HPO), and Synthetic Minority Oversampling (SMOTE). In [11] authors compared hybrid ensemble and deep learning methods and created a Champion-challenger structure.

The authors of the article [12] offer an innovative solution for streaming data processing Stream Cube based on an incremental calculation scheme and polynomial decomposition of metrics. To achieve the goal of intelligent decision making, they also create a real-time intelligent data processing system and an AI model for analysis.

Machine learning can be generally categorized as either supervised learning methods or unsupervised learning methods [13]. Supervised learning focuses on using knowledge of known classifications in order to create an optimal algorithm for predicting future classifications.

Unsupervised machine learning, however, evaluates the similarity between and among different variables for purposes of finding special or interesting patterns, including latent groups or clusters, embedded in the data. Cluster analysis is the most commonly used algorithm in unsupervised machine learning. For many clustering algorithms (e.g., K-means, agglomerative clustering), the number of clusters has to be determined before learning. To avoid having to identify the number of clusters, Caron, Bojanowski, Joulin, and Douze [14] proposed a deep clustering unsupervised learning method for which the stopping rule is not based on a predetermined number of clusters. Instead, the Caron, et al. method iteratively learns the features of the latent groups with a standard clustering algorithm until an accuracy criterion is satisfied (e.g., all clusters are sufficiently homogeneous). A nice description of several supervised and unsupervised machine-learning methods that have potential for application to test security was provided by Man, Harring, and Sinharay [15].

Taking into account the specifics of the system being developed, the starting point of making it is a data collection, and the main sources of data are information about the user, transaction data, etc. The most challenging is that the messages must be processed in a near real-time. Thus, big data technologies will be applied [16], which entails existing problems:

Ensuring of data storage integrity, availability and confidentiality;

- The complexity of structure, sorting and distribution when making samples and searching for a specific element from the general system;
- Low processing speed (in comparison with the amount of data), which can lead to a long waiting time for a response when searching for a specific position, as well as their obsolescence during the process of processing;
- Lack of effective processing algorithms that take into account the amount of data storage, data structure and search methods for the required element;
- A large amount of noise and the process of accounting for them when working with datasets.

The following two articles were chosen as the most promising concerning introducing of new approaches:

1) In the article" An Optimized Quantitative Argumentation Debate Model for Fraud Detection in E-Commerce Transactions" [17] a method for detecting of a fraud based on quantitative argumentation is considered. An argument tree Fig. 1 is built by combining human knowledge and knowledge gained from data. The existing Quantitative Debate Argument (QuAD) is expanded by adding the strength of the correlation between arguments and the Particle Swarm Optimization (PSO) algorithm is used to determine the correlation of strength between the arguments. The processing is based on creating an argument tree for fraud detection based with an expert knowledge.

2) Another promising approach is to use a scalable big data ecosystem module based on standard Apache tools such as Kafka, Spark and Cassandra [18]. The main advantage of these components is that they handle fault tolerance and task distribution in the same way. To collect transactions across distributed queuing systems, Kafka is used, for data analysis Apache Spark with a Map-Reduce implementation that automatically distributes calculations between assigned resources and combines the results in a distributed file system. The

proposed structure Fig. 2 relies on Spark Streaming, which processes the data stream in mini-packets. Spark performs three tasks in the pipeline: aggregating previous transactions to perform functional engineering, online classification of transactions, which returns the perceived risk of fraud.

B. Fraudulent transactions detection

As was mentioned earlier, the task of introducing machine learning into anti-fraud systems is a topical subject now. However, the most important goal is to accurately classify a transaction in real-time or near real-time with minimal involvement of bank security staff. A generalized scheme for detecting fraudulent transactions is as follows Fig. 3 [19].

Fig. 1. Argument tree for fraud detection

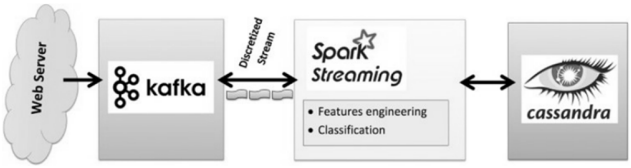

Fig. 2. Big data architecture

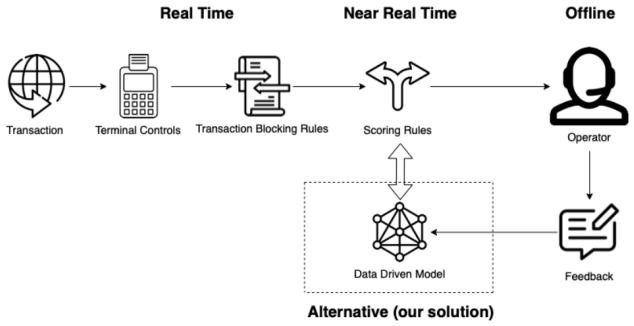

Fig. 3. A generalized scheme for detecting fraudulent transactions

Initial transaction verification is implemented through Terminal controls and Transaction Blocking Rules. During the Terminal controls phase, a technical check of the card and the personal account is performed (for example, incorrect PIN or insufficient funds in the account). Transaction Blocking Rules are a set of simple rules that are designed to detect obvious cases of fraud (purchase from abroad). Transactions which don't pass the Transaction Blocking Rules are flagged as suspicious and sent for delayed verification by an operator-analyst. The initial verification takes place in real time and covers a huge number of transactions, so at this stage there is no way of taking into account the behavior of each individual bank card user and identifying unusual fraudulent behavior. Transactions that do not pass the initial verification are automatically canceled.

After successfully passing the initial check, the transactions go to the additional check, which is performed by Scoring Rules and the Data Driven Model module in modern bank anti-fraud systems. Scoring Rules are flexible guidelines based on personal data, aimed at detecting more specific cases of fraud (large atypical purchases for a particular user). The Data Driven Model module is based on Big Data and machine learning and performs a more detailed check of transactions to determine their class. Additional checking is done in near real time, that is, the transaction is already done but not yet authorized (validated). At each step of the additional validation, the transaction is given a score. Based on the average of the scores, the transaction may be deemed suspicious and sent for delayed verification by the operator-analyst. Transactions that successfully pass the primary and additional verification are marked as legitimate and confirmed in the system.

The last step in detecting fraudulent transactions is a delayed check by an operator-analyst, who sees alerts from all of the above systems (except Terminal controls). It is the operator-analyst who ultimately decides on each suspicious transaction, namely which ones are legitimate and which are fraudulent and can either approve or cancel the transaction accordingly. The Data Driven Model is supposed to be trained (improved) based on Feedback from the operator-analyst in order to reduce the number of pending alerts [20].

The user of banking services should not feel server processing delays when fraud is detected during a transaction under perfect conditions. However, this is only possible if the transaction is clearly defined as fraud, e.g., very different from normal user transactions. Transactions which cannot be clearly defined and have a high probability of being illegitimate are sent for delayed verification by a specialist-analyst. This significantly reduces the response time, but provides an opportunity not only to reduce to zero system errors as a result of false positive and false negative transactions, but also to detect cases of economic crimes such as tax evasion and terrorist financing. Thus, the following requirements are imposed on a modern anti-fraud system:

- fast processing speed (good performance);
- a high percentage of correctly identified transactions;
- a low percentage of suspicious transactions at the last (pending) stage, and as a result a lower involvement of specialists in the process of detecting fraudulent transactions.

As a result, the task of the proposed software product is to automate routine tasks of determining the status of transactions for ordinary users. Tasks that involve complex and ambiguous types of financial fraud have slightly different directions for solution - namely, the analysis of user behavior and history of his payments. To solve this optimization problem, the following software architecture is proposed Fig. 4:

1. Dataset of remote banking transactions.
2. Kafka to store and distribute samples of input data about banking transactions.
3. The transaction data are processed by PyFlink using a Python script with trained Machine Learning models.
4. Elasticsearch is used to store results and provide an efficient query service.
5. Kibana is an open-source data visualization dashboard for Elasticsearch that uses the visualization of the PyFlink pipeline through the control panel.

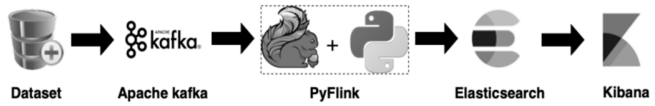

Fig. 4. Proposed fraud detection architecture for big data

C. Data

1) Dataset Specification: To conduct an experiment the dataset of banking transactions from Kaggle data science community was used [21]. It has a volume of 502 MB and consists of 1,852,394 transactions, of which 9,651 are fraudulent transactions. Each transaction is described by 22 parameters, including: the coordinates of the bank card holder and the merchant, name of the merchant, transaction amount, its category and exact completion time. It is also including personal data of the transaction initiator such as name, gender, date of birth and professional affiliation. The main characteristics of the dataset are:

- Total amount of transactions: 1,852, 394
- Fraudulent transactions: 9,651
- Total number of predictors (columns): 22
- Data volume: 502 MB
- Year: 2020

2) Dataset Generation: It is necessary to mention that the dataset does not contain real transactions because it was generated using Sparkov Data Generation which is a GitHub tool created by Brandon Harris [22]. The key component of data generation algorithm is the Python library called" Faker", it utilizes predefined list of merchants, customers and payment categories to create transactions. This tool is trying to simulate a behavior of a real persons based on special profiles that include the following parameters: minimum and maximum number of transactions per day, their distribution by time, days of week, season and payment categories.

Below are some examples of user profiles:

- adult urban women at age from 25 to 50;
- adult rural men older than 50;
- young female adults.

Author of the dataset used all available profiles to generate transactions and then merged the results together to obtain a more realistic representation of simulated transactions.

3) Data Preprocessing: To conduct an experiment and obtain a reliable result, the data was preprocessed and translated to such form which would be acceptable for previously selected machine learning algorithms. The main data preprocessing stages are displayed on "Fig. 5".

At first, it was necessary to extract such predictors from the dataset which have a statistical relationship with" is fraud" label. So, these variables would be the most useful for training machine learning classifiers. In order to determine the relationship between the variables, the correlation function was calculated using corr() function for Pandas data frame. The correlation function results are displayed in Table I. According to the results, the most valuable variables for classification are: amount, Unix time, and geographic location of card holder and merchant.

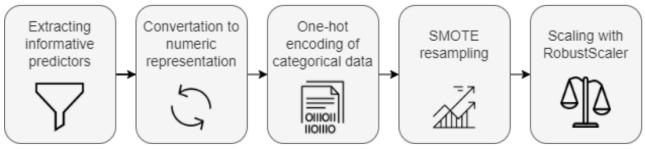

Fig. 5. Data preprocessing stages

Then the gender variable was converted to numeric format by replacing" M" and" F" letters to zeroes and ones.

Furthermore, it was obvious that the payment category has some relation with target variable, so it was also translated to numeric representation using one-hot encoding. It is a type of data encoding that represents each category as sequence of bits when the bit for corresponding category is equal to one while other categories' bits are zeroes. For example, if RGB colors is one-hot encoded, blue color corresponds to 001-bit sequence. Thus, this preprocessing technique helps to pass categorical data to machine learning algorithm for analysis.

The dataset specification makes it clear that it is highly unbalanced. It means that a positive class is represented by much fewer elements than a negative one. This fact may lead to poor classification performance for positive class. So, it was required to expand examples of minority class. Perhaps the most widely used approach to synthesizing new examples is called the Synthetic Minority Oversampling Technique, or SMOTE for short. This technique was described by Nitesh Chawla, et al. in their 2002 paper [23].

SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.

TABLE I. CORRELATION OF NUMERIC PREDICTORS WITH" IS_FRAUD" LABEL

Predictor	Pearson correlation
Card number	-0.001125
Amount	**0.209308**
Gender	**0.005844**
Zip code	-0.002190
Latitude	**0.002904**
Longitude	**0.001022**
City population	0.000325
Unix time	**-0.013329**
Merchant latitude	**0.002778**
Merchant longitude	**0.000999**

Some input variables frequently have very large values relative to the other input variables, these values can dominate or skew some machine learning algorithms. In order to eliminate this issue and remove outliers robust scaling was applied. The values of each variable have their median subtracted and are divided by the interquartile range (IQR) which is the difference between the 75th and 25th percentiles.

$$value = \frac{value - median}{p(75) - p(25)} \quad (1)$$

D. Methods

During the study, the models were trained using well-known machine learning methods. The methods differ in mathematical formulas, work approaches and computational requirements. Therefore, the task of identifying the most optimal method is one of the main tasks to optimize the learning process and the speed of the system's response. After preprocessing the datasets, supervised and ensemble learning methods were used, implemented in production ready libraries Scikit Learn (Sklearn) library. It is a popular and widely used Python package for working with Data Science and Machine Learning.

During the experiment, the following methods were used:

1) Logistic Regression (LR): allows to determine the belonging of a testing dataset to a class based on the determination of the conditional probability identified from the training dataset. The principle of operation consists in the boundary division of space into areas corresponding to the classes "0" - legitimate transactions and "1" - fraudulent transactions. Implemented with *LogisticRegression* from *sklearn.linear_model library*.

2) K-Nearest Neighbors (KNN): defines the class of a testing dataset based on its closest neighbors. To determine the degree of proximity, the smallest distance to neighboring objects is used, calculated through the Euclidean metric, Manhattan distance, Chebyshev's metric. The choice of the number of neighbors K taken into account remains at the mercy of the researcher. [24] Implemented with *KNeighborsClassifier* from *sklearn.neighbors library*.

3) Gaussian Naive Bayes (GNB): the classifier is constructed using the Bayess theorem, which allows one to determine the probability of the outcome of an event, provided that something else happened in the presence of statistical interdependence. In practice, the maximum likelihood method is often used to estimate the parameters to create a statistical model based on the sample and provides an estimate of the parameter. Implemented with *GaussianNB* from *sklearn.naive-bayes library*.

4) Multilayer Perceptron (MLP): it is a classic type of feedforward neural network. Consists of three main layers: input, hidden, output. All neurons, except the input neurons, contain a nonlinear activation function. The training process uses an error backpropagation algorithm that trains all layers. The main application is the construction of regression models and classifiers. Implemented with *MLPClassifier* from *sklearn.neural_network library*.

5) Decision Tree (DT): on the basis of training data it makes classes of objects using conditional construction and branching. For this, a boundary indicator (the probability of attribution to a class) is formed for the inclusion of a test object in one of the sets until the probability of assignment of the object will not unambiguously determine the class. Implemented with *DecisionTreeClassifier* from *sklearn.tree library*.

During the process of machine learning methods, classification and regression errors often occur, which entail incorrect predictions and ambiguous situations. Since there are no best algorithms, and the quality of their work depends on the training sample, then under certain conditions, weak algorithms appear, the prediction of which is only slightly better from a random distribution. In such a situation, in order to obtain the best samples, combination technologies are used: methods, samples, retraining. Such methods are called ensemble methods. The use of ensembles in the research process can solve the problem of optimizing the learning process and reducing the response time. The principles of operation of ensemble methods can be generalized into some categories:

- Usage of several machine learning methods combined in a single model in order to summarize their advantages with the possibility of mutual elimination of disadvantages when working with data.
- Repeated training of the model by various methods in order to obtain the highest quantitative estimates of the system. Allows you to handle uncertainties and fight classification errors

However, when using these approaches, it is worthwhile to carefully think about the constituent parts of the and over the sequence of actions during training - so as not to get the opposite of the expected results, to prevent overtraining or wasteful use of computational resources. In the research work at the stage of training models, the following ensemble methods were used

6) Random Forest (RF): a machine learning algorithm based on an ensemble of decision trees, as when used together, it shows better results than a single tree. Classification of objects is carried out by voting - each tree assigns an object to one of the classes. As a result, the object is assigned the class that has been voted for by the most trees. Implemented with *RandomForestClassifier* from *sklearn.ensemble* library.

7) Adaptive Boosting (AdaBoost): the property of adaptability is expressed in the construction of the following classifier based on ambiguous and misclassified objects in the past. During each subsequent iteration, their weights increase, due to which the next classifier concentrates more of his attention on them. The combination makes it resistant to overfitting, but sensitive to outliers. Implemented with *AdaBoostClassifier* from *sklearn.ensemble* library.

8) Gradient Boosting (GradBoost): is a machine learning technique for regression, classification and other tasks, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differential loss function. Implemented with *GradientBoostingClassifier* from *sklearn.ensemble* library.

9) eXtreme Gradient Boosting (XGBoost): is an advanced gradient boosting approach, with the ability to parallelize, making it much faster, reducing overfitting, and increasing performance. Optimization achieved through the use of cross validation in iterations, regularization, the ability to work with sparse data, and the ability to finetune the model. The method shows good results when working on the prediction process. Implemented with *XGBClassifier* from *xgboost* library.

10) Bootstrap Aggregating (Bagging): the essence lies in dividing the general sample X into a set of sub samples on which the algorithm is trained. From the results obtained, the averaged value of the statistical parameters is extracted. This reduces the amount of variance and standard deviation, which in turn minimizes over fitting and compensates for the errors in system. Implemented with *sklearn.ensemble* from *sklearn.ensemble* library.

E. Results

The experiment was conducted on the computing power of Google Collab with 12.69 GB RAM and 2 cores.

1) Evaluation metrics: One of the key concepts in evaluating the quality of a trained model is the confusion matrix, which is a tabular representation of the predictions made by the model. The rows of this table represent the classes predicted by the algorithm, and the columns represent the real classes. The values of the cells of this matrix are used to calculate all the estimated metrics.

Accurately predicted objects of positive and negative classes are called True Positives (TP) and True Negatives (TN), respectively. Errors of classifying a positive class as negative are called False Negatives (FN), and negative class as positive are called False Positives (FP). Also in mathematical statistics, FP errors are called errors of the first kind, and FN cases are called errors of the second kind.

The following metrics were used in order to evaluate the quality indicators of the proposed research development - an anti-fraud system based on machine learning:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

1) Precision, or positive predictive value, is the proportion of positive outcomes that were correctly classified to the total number of positive predicted outcomes.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

2) Recall or sensitivity is otherwise known as the true positive rate (TPR). TPR is the number of actual positives that are predicted to be positive.

3) F_1 measure represents the harmonic average of accuracy and completeness. The value varies from 0 to 1, if the value is high, the F_1 measure indicates high classification efficiency.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (4)$$

4) Area under the curve (AUC). Represents the ability to distinguish between classes. When the AUC is higher, the predictive ability is better, the value is in the range of 0 to 1.

$$AUC = \frac{1 + TPR - FPR}{2} \quad (5)$$

Fall-out, also FPR (false positive rate), shows the share of incorrect responses of the classifier to the total number of objects outside the class. In other words, how often the classifier makes an error when assigning an object to a class.

$$FPR = \frac{FP}{TN + FP} \quad (6)$$

2) Experimental results: The experiment results show (Table II) that the ensemble methods have the highest Precision indicators, since when they are used, repeated training occurs on the erroneous predictions of the previous algorithm. For the first (reduced) dataset the AUC is the highest for the AdaBoost, XGBoost and MLP methods. The graphical interpretation of ROC curves for the experiment is presented in the Fig. 6.

TABLE II. CLASSIFICATION RESULTS ON A REDUCED DATASET

Classifier	Precision	Recall	F_1	AUC
LR	*0.10*	*0.77*	*0.17*	*0.907*
KNN	*0.62*	*0.71*	*0.66*	*0.881*
GNB	*0.02*	*0.80*	*0.04*	*0.839*
DT	*0.68*	*0.76*	*0.71*	*0.879*
MLP	***0.58***	***0.82***	***0.68***	***0.983***
RF	*0.87*	*0.73*	*0.79*	*0.975*
AdaBoost	***0.89***	***0.80***	***0.85***	***0.987***
GradBoost	***0.78***	***0.80***	***0.79***	***0.982***
Bagging	*0.77*	*0.76*	*0.77*	*0.956*
XGBoost	***0.73***	***0.81***	***0.76***	***0.984***

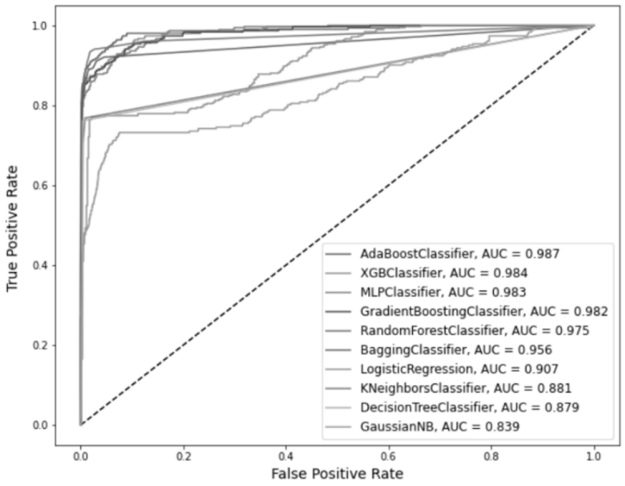

Fig. 6. Reduced dataset ROC results

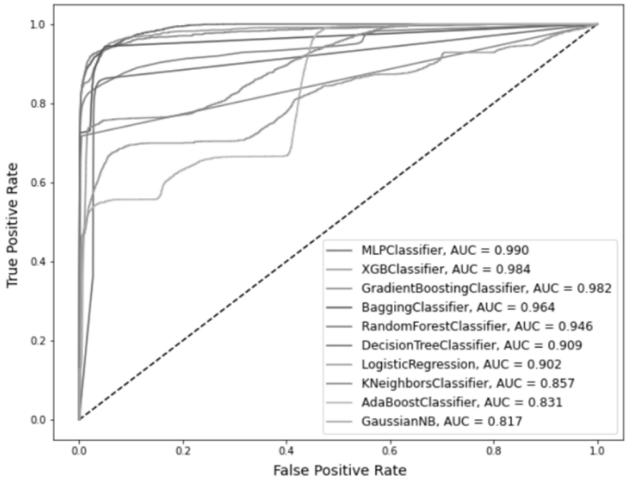

Fig. 7. Whole dataset ROC results

An experiment on a complete dataset practically reproduces the conditions for working with a Big Data. Processing of a large transactions amount has significant differences concerning detection accuracy and performance. In this case, ensemble methods also showed good results. It can be observed that the Random Forest algorithm has the highest $F1$ score. However, for the AUC, the MLP method (AUC = 0.990, Recall = 0.83) turned out to be the best algorithm in the terms of fraud detection, since when working with a bigdata, it is important not to overfit the model. For MLP, this is achieved by the fact that training is performed not to minimize the error, but to stabilize the network weights, together with the backpropagation algorithm. The results of the experiment for the whole dataset are given in Table III. A graphical interpretation of the error matrix of the experiment performed in the form of a ROC curve can be seen in Fig. 7.

Comparing the detection metrics (AUC, F_1) of the methods with the indicators of their performance (training time and classification time), it can be found that ensemble methods take a longer time to learn due to the multilevel implementation of the algorithm. As for the classification time they have better results, however still are unable to ensure the operation of the system in a real-time mode. To ensure a quick response to a payment system client, a minimum classification time is required. During an experiment, it was shown that the Logistic Regression and Decision Tree methods are faster. As a result, application of these methods will increase performance but decrease the accuracy of predictions, leading to misclassifications. The K-Nearest Neighbors method showed a good indicator of training time; however, the longer prediction time makes the method weakly suitable for working in a real-time. Comparative characteristics of the whole metrics and the running time of the algorithm are presented in Table IV. Timing charts for the methods are shown in Fig. 8 and Fig. 9.

TABLE III. CLASSIFICATION RESULTS ON A WHOLE DATASET

Classifier	**Precision**	**Recall**	***F1***	**AUC**
LR	*0.07*	*0.76*	*0.13*	*0.902*
KNN	*0.58*	*0.65*	*0.61*	*0.857*
GNB	*0.01*	*0.71*	*0.02*	*0.817*
DT	*0.57*	*0.78*	*0.66*	*0.909*
MLP	***0.50***	***0.83***	***0.62***	***0.990***
RF	*0.81*	*0.70*	*0.75*	*0.946*
AdaBoost	*0.60*	*0.82*	*0.69*	*0.831*
GradBoost	***0.60***	***0.81***	***0.69***	***0.982***
Bagging	*0.74*	*0.72*	*0.73*	*0.964*
XGBoost	***0.59***	***0.82***	***0.69***	***0.984***

TABLE IV. PERFORMANCE OF TRAINING AND CLASSIFICATION BY TIME

Classifier	**AUC**	**Training (s)**	***Classification (s)***
LR	*0.902*	*191*	*0.02*
KNN	*0.857*	*34*	*305.00*
GNB	*0.817*	*1*	*0.20*
DT	*0.909*	*73*	*0.08*
MLP	***0.990***	***674***	***0.32***
RF	*0.946*	*586*	*5.01*
AdaBoost	*0.831*	*2255*	*11.80*
GradBoost	*0.982*	*3204*	*1.48*
Bagging	*0.964*	*5162*	*11.40*
XGBoost	***0.984***	***667***	***2.05***

Thus, according to the results of the experiment, the following conclusions can be done:

1) There is no method that will show the best results, according to all the necessary indicators.
2) Ensemble methods have the best detection quality, but their underlying multilevel learning algorithms decrease the classification and training time.
3) Despite the high accuracy, not all ensemble methods are suitable for the real-time system. The methods have good potential for work in the process of fraud detection; however, if it is necessary to reduce the response time, they are inferior to classical methods. The best-weighted results are shown by the MLP algorithm, which demonstrated an appropriate classification time and high detection accuracy.
4) With an increase in the volume of the dataset and approaching indicators comparable to big data, such as volume, velocity and value, the performance can vary.

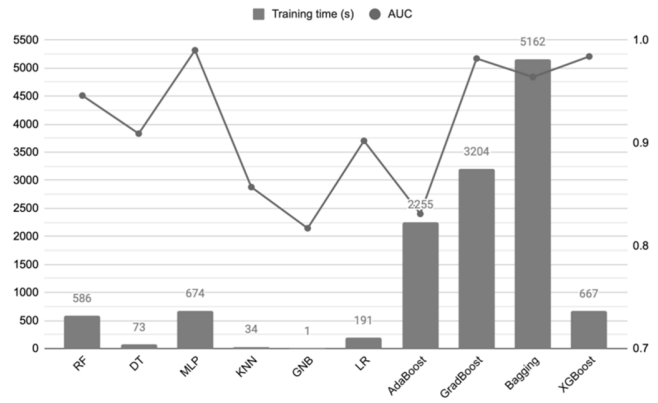

Fig. 8. Training time of ML algorithms

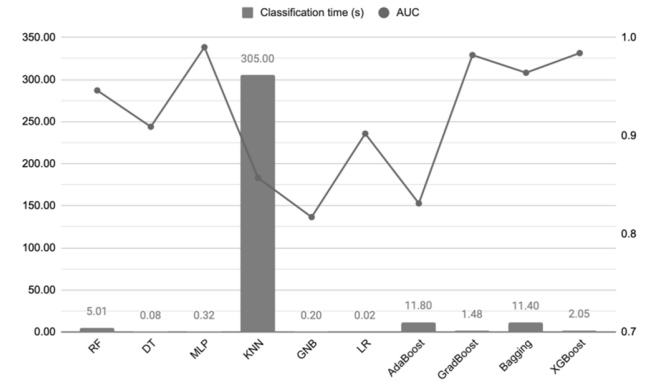

Fig. 9. Classification time of ML algorithms

III. CONCLUSION

In the course of the work, a review of existing approaches to detecting fraudulent transactions was conducted and a generalized scheme of this process was presented. The first step in the implementation of this software is to identify the most optimal machine learning methods, which was the main goal of this paper. To achieve this goal, the following tasks were solved. First of all, an architecture for detecting fraudulent transactions in the Big Data ecosystem was proposed. An impersonal dataset containing sets of bank transactions and marked into fraud/non-fraud classes was chosen. This dataset has been cleaned of empty values and unbalanced classes have been removed. To conduct an experiment and obtain reliable results, the data was reprocessed. Then as a result of comparative analysis the most promising ML methods (including ensemble methods) were selected in terms of classification and training performance, accuracy and detection recall. As a result of the experiment on a dataset containing 1,852,394 records, the methods that showed themselves the best in terms of performance were identified. The best weighted results are shown by the MLP algorithm, which demonstrated an appropriate classification time and high detection accuracy. The next stage in the implementation of the proposed software architecture will be to evaluate their performance in the large Big Data ecosystem.

IV. ACKNOWLEDGMENT

The main difficulty in comparing the presented results with the achievements of other researchers is the use Fig. 8: Training time of ML algorithms Fig. 9: Classification time of ML algorithms of various datasets for testing models. Therefore, according to the presented results, one can only draw an analogy with respect to the machine learning methods being tested. The article [25] compares 3 algorithms such as Logistic regression, Naive Bayes, K-nearest neighbor. Evaluating the main indicator AUC, a similar trend is seen in the order of increase in the value of this metric, 0.918, 0.829, 0.633, respectively. Thus, the presented article achieves better results by examining more algorithms.

REFERENCES

[1] C. Lynch, How do your data grow?, Nature 455 (2008) 28–29.
[2] A. A. Menshchikov, M. Y. Fedosenko, Methods and approaches to data preprocessing under the condition of a strong imbalance of classes, Student scientific and educational journal ”StudNet” 9 (2021) 1–15. doi:10.24411/2658- 4964- 2021- 103682.
[3] Z. Li, H. Zhang, M. Masum, H. Shahriar, H. Haddad, Cyber fraud prediction with supervised machine learning techniques, Proceedings of the 2020 ACM Southeast Conference (2020) 176–180. doi:https: //doi.org/10.1145/3374135.3385296.
[4] M. Sapozhnikova, A. Nikonov, A. Vulfin, M. Gayanova, K. Mironov, D. Kurennov, Anti-fraud system on the basis of data mining technologies, IEEE International Symposium on Signal Processing and Information Technology (ISSPIT) (2017) 243–248. doi:10.1109/ISSPIT.2017.8388649.
[5] Y. Abakarim, M. Lahby, A. Attioui, An efficient real time model for credit card fraud detection based on deep learning, SITA’18, Association for Computing Machinery, New York, NY, USA, 2018. URL: https://doi.org/10.1145/3289402.3289530. doi:10.1145/3289402.3289530.
[6] D. Prusti, D. Das, S. K. Rath, Credit card fraud detection technique by applying graph database model, Arabian Journal for Science and Engineering 46 (2021). doi:10.1007/s13369- 021- 05682- 9.
[7] C. Ikeda, K. Ouazzane, Q. Yu, A new framework of feature engineering for machine learning in financial fraud detection (2020).
[8] A. Thennakoon, C. Bhagyani, S. Premadasa, S. Mihiranga, N. Kuruwitaarachchi, Real-time credit card fraud detection using machine learning, in: 2019 9th International Conference on Cloud Computing, Data Science Engineering (Confluence), 2019, pp. 488–493. doi:10.1109/CONFLUENCE.2019. 8776942.
[9] L. Zhou, S. Pan, J. Wang, A. V. Vasilakos, Machine learning on big data: Opportunities and challenges, Neurocomputing 237 (2017) 350–361. URL: https://www.sciencedirect.com/ science/article/pii/S0925231217300577. doi:https: //doi.org/10.1016/j.neucom.2017.01.026.
[10] N. Rtayli, N. Enneya, Enhanced credit card fraud detection based on svm-recursive feature elimination and hyper-parameters optimization,

Journal of Information Security and Applications 55 (2020) 102596. URL: https://www.sciencedirect.com/science/article/pii/S221421262030764X. doi:https://doi.org/10.1016/j.jisa.2020.102596.

[11] E. Kim, J. Lee, H. Shin, H. Yang, S. Cho, S. kwan Nam, Y. Song, J. a Yoon, J. il Kim, Champion- challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning, Expert Systems with Applications 128 (2019) 214–224. URL: https://www.sciencedirect.com/science/article/pii/S0957417419302167. doi:https://doi.org/10.1016/j.eswa.2019.03.042.

[12] T. Zheng, G. Chen, X. Wang, C. Chen, X. Wang, S. Luo, Real-time intelligent big data processing: technology, platform, and applications, Science China Information Sciences 62 (2019). doi:https://doi.org/10.1016/j.inffus.2017.09.005.

[13] Y. Pan, J. A. Wollack, An unsupervised-learningbased approach to compromised items detection, Journal of Educational Measurement 58 (2021) 413–433. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/jedm.12299. doi:https://doi.org/10. 1111/jedm.12299.

[14] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[15] K. Man, J. R. Harring, S. Sinharay, Use of data mining methods to detect test fraud, Journal of Educational Measurement 56 (2019) 251–279.

[16] N. Wang, Y. Liu, Z. Liu, X. Huang, Application of artificial intelligence and big data in modern financial management, in: 2020 International Conference on Artificial Intelligence and Education (ICAIE), 2020, pp. 85–87. doi:10.1109/ICAIE50891.2020.00027.

[17] P. Chi, Y. Lu, B. Liao, L. Xu, Y. Liu, An optimized quantitative argumentation debate model for fraud detection in e-commerce transactions 36 (2021) 52–63. doi:10.1109/MIS.2021.3071751.

[18] F. Carcillo, A. Dal Pozzolo, Y.-A. Le Borgne, O. Caelen, Y. Mazzer, G. Bontempi, Scarff: A scalable framework for streaming credit card fraud detection with spark, Information Fusion 41 (2018) 182–194. URL: https://www.sciencedirect.com/science/article/pii/S1566253517305444. doi:https://doi.org/10.1016/j.inffus.2017.09.005.

[19] A. D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, G. Bontempi, Credit card fraud detection: a realistic modeling and a novel learning strat- egy, IEEE Transactions on Neural Networks and Learning Systems (2017).

[20] Y. Jianhao, Design and implementation of bank wind control anti-fraud project based on big data technology, Journal of Physics: Conference Series 7 (2019) 1–7. doi:10.1088/1742- 6596/1345/ 2/022064.

[21] K. Shenoy, B. Harris, Credit card transactions fraud detection dataset, 2020. URL: https://www.kaggle. com/kartik2112/fraud-detection.

[22] B. Harris, J. Plotkin, Sparkov data generation, 2020. URL: https://github.com/namebrandon/Sparkov_ Data_Generation.

[23] N. V. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, Smote: Synthetic minority over-sampling technique, Journal Of Artificial Intelligence Research 16 (2002) 321–357. doi:10.1613/jair.953".

[24] V. R. Ganji, S. N. P. Mannem, Credit card fraud detection using anti-k nearest neighbor algorithm, International Journal on Computer Science and Engineering (IJCSE) 4 (2012) 1035–1039.

[25] F. Itoo, S. Singh, et al., Comparison and analysis of logistic regression, naïve bayes and knn machine learning algorithms for credit card fraud detection, International Journal of Information Technology 13 (2021) 1503–1511.