Grey Wolf Optimizer for feature selection. Example of transaction fraud detection.

Alexander Agafonov
Innopolis University
Innopolis, Russia
a.agafonov@innopolis.university

Polina Bazhenova
Innopolis University
Innopolis, Russia
p.bazhenova@innopolis.university

Daria Lebedeva
Innopolis University
Innopolis, Russia
d.lebedeva@innopolis.university

Abstract—**Grey Wolf Optimizer (GWO) is a population-based optimization algorithm inspired by the social hierarchy and hunting behavior of grey wolves. The algorithm has been successfully applied in various optimization problems and has shown better performance than other metaheuristic algorithms. This work observes its performance as a feature selectiong technique for transaction fraud detection. The results showed that GWO does not increase performance for the specified task, but still reduces the number of features without the loss in metrics.**

Index Terms—**Grey Wolf Optimizer, feature selection, transaction fraud detection**

I. Introduction

While creating the machine learning model, one of the steps the scientist encounters is feature selection. Feature selection is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Most probably, the datasets are set up from a huge number of columns some of which appear to be irrelevant or redundant that they can easily be removed without much loss of information. It appears to be a complicated task for selecting them by hand. To overcome this issue, a number of techniques were developed that are quite common nowadays. [1] describes a variety of such methods to select a subset of variables to take into our model. The aim of this work is to observe the Grey Wolf Optimization nature-inspired algorithm as one more feature selection technique. The work will be done on the base of transaction fraud detection. Transaction fraud is a common risk for businesses that take online payments. Fraudsters may use lost or stolen credit card data to make purchases from strangers' accounts. The goal of a model is to detect these illegal transactions with the highest accuracy.

II. Related Work

Various studies have been performed on the topic of transaction fraud detection. [2] compares the performance of Random Forest, Logistic Regression, KNN, Support Vector Machine, Decision Tree, and Naive Bayes classifiers. [3] reviews works on using Neural Networks, Logistic Regression, Decision Trees, Genetic Algorithms, Clustering Techniques, and Outlier Detection algorithms but without evaluating their performances. [4] observed the performance of Isolation Forest and Local Outlier Factor methods to find fraud transactions. In [6] genetic algorithms are used for the classification of transactions.

III. Methodology

A. Dataset

The dataset that was used was taken from the IEEE-CIS Fraud Detection competition on Kaggle. It was provided by Vesta Corp. which specializes in guaranteed e-commerce payments. The dataset can be found by the link: https://www.kaggle.com/competitions/ieee-fraud-detection/data. This dataset contains two tables: transaction and identity which are linked through the TransactionId field. The first table contains 393 features and 590 040 records in the train set. The second one has 41 columns and 144 233 records. After joining two datasets on the transactionId field, we had 590 040 records with 434 features The dataset contains the target column isFraud which is about to be predicted. the dataset is far from balanced. Only 3,5 % of the records are fraud.

B. Model Selection

There is a number of machine learning models that are used for classification problems: Decision trees, Random Forest, KNN, SVM, Naive Bayes, and Logistic Regression. Although Artificial Neural Networks can be used for classification as well and perform very well [7], they don't fit our aim for evaluating feature selection techniques because they can decide the features they need as they are being trained. Besides, it takes much more time to train them on large datasets. Talking about KNN, we should mention that this method does not require training but needs to store all of the data to work. So, we decided not to use it as well. [2] shows the great performance of Random Forest, Logistic Regression, and Decision Trees on transaction fraud detection. Though, Decision Tree performs slightly worse than the other two methods. [7] shows the opposite result: Logistic Regression had a little more accuracy loss than Decision Tree but much worse precision. [8] compared Logistic Regression, Random Forest, and Decision Tree models. The second one performed better than the other ones, but Decision Tree still had good results with about 1% less score while Logistic Regression had a smaller score by 5%. After analyzing mentioned works, we decided to use the Decision Tree classifier for our experiment. Moreover, Decision Trees give easily interpretable results so the workers can see and analyze how the algorithm makes the decision.

C. Preprocessing

Firstly, we joined two original datasets in one by the TransactionId field. Then, we dropped all the features that contained only null values. We observed 40 such columns. Also, we dropped TransactionId field itself due to its uselesness. Further, we imputed all missing data with most frequent values. Then, we splitted the data into train and test set. We took test set size as 20% of the dataset length. As the dataset is very unbalanced, we undersampled the training set. We randomly dropped records that are not fraud so that we ended with the training set with fraud / not fraud proportion as 50/50. After these steps we had a training dataset with 392 features and 16060 rows.

D. Grey Wolf Optimizer

The core of the work was to observe the performance of the Grey Wolf Optimizer nature-inspired algorithm. The technique was described in [5]. The method uses the process of how the pack of wolves searches and hunts for prey. The pack has a special hierarchy of species: division for alpha, beta, delta, and sigma ones. In the process of decision making the whole pack relies on the actions of wolves that stay higher in the hierarchy. The algorithm uses the same approach: the whole population of vectors moves on the basis of three ones with the best score so far. Since the algorithm itself was not changed anyhow, the problem that was need to be solved is the representation of the specie. Suppose, after the dataset was preprocessed it consisted of n features. Thus, we represented the specie as an n - dimensional vector. Each number of the vector uniquely mapped the feature in the dataset and was used to decide whether this feature should be dropped or fitted into the model. These numbers belonged to the $(0, 1)$ real-valued interval. If the value exceeded the specified threshold (e.g. 0.5), the feature was kept. Otherwise, it was dropped. This representation is shown in “Fig. 1”.

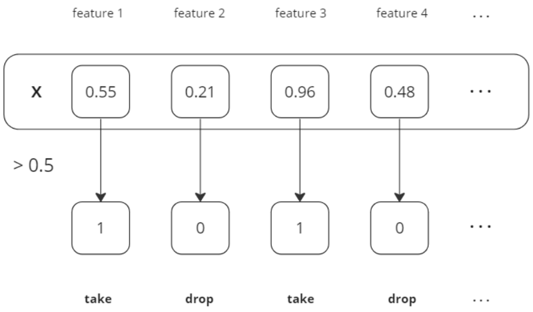

Fig. 1. Specie representation.

E. Algorithm

The whole pipeline is quite simple. Given the dataset, it was firstly preprocessed (Section III-C). Then the Grey Wolf Optimization algorithm was created and the population was randomly initialized. The population consisted of n - dimensional vectors. Then the iteration process was started. Each iteration consisted of several steps. Firstly, for each individual specie, the feature subset was selected as was discussed in Section III-D. Then, categorical features of this subset were one-hot encoded and fitted into the model. The cross-validation scores were observed for individuals and the top three ones were selected as alpha, beta, and delta species. Finally, the whole population was updated according to the Grey Wolf technique and the algorithm proceeded to the next iteration. On each iteration, alpha scores were kept in the array for further evaluation of the performance. After the optimizer finished, the top species was selected, and its according features were selected, encoded, and fitted into the final model. Then, the performance was evaluated on the test set. The algorithm is shown in “Fig. 2”.

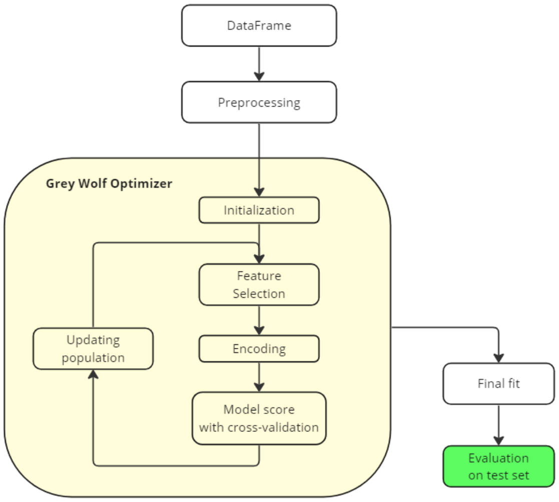

Fig. 2. Algorithm scheme.

F. Compared techniques and metrics

We compared the results of models trained on subsets of features that were evaluated by different algorithms. These are Grey Wolf Optimizer and Recursive Feature Elimination (RFE). The first one is the item of our research. The Recursive Feature Elimination’s goal is to select features by recursively considering smaller and smaller sets of features based on the feature importance scores yielded by the model. The metrics we compared were the balanced accuracy, recall, precision, and F1 scores evaluated on the test set, and the time it took for the feature selection algorithm to run.

IV. Experiments and Evaluation

We ran our GWO algorithm for two population sizes (10 and 30). During the algorithm runs, we saved the best cross-validation score over the population and the number of features according to this fit. “Fig. 3” and “Fig. 4” show the recored CV - scores for two population sizes. “Fig. 5” and “Fig. 6” show the number of features. Also, we created “Table I” that shows the metrics and “Table. II” that shows running times for compared techniques. “Fig. 7” pics the results from tables in graphical form

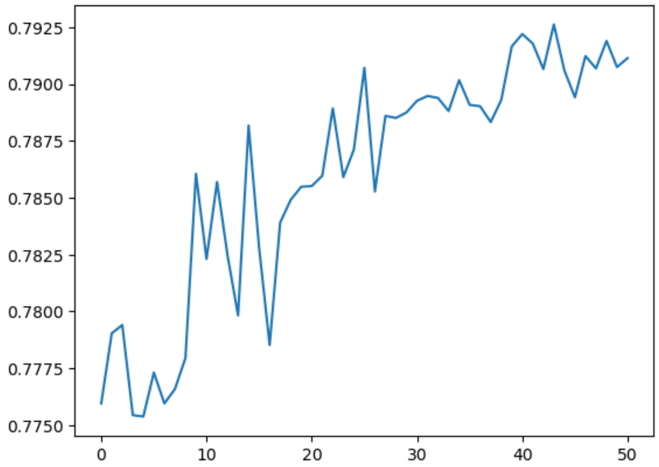

Fig. 3. CV - score at each iteration for $population\_size = 10$.

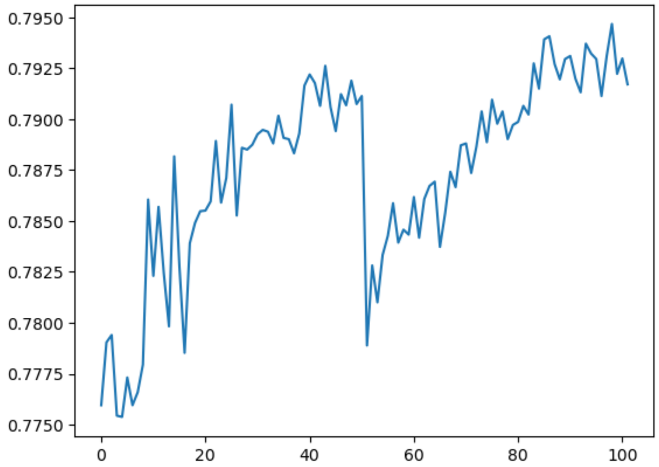

Fig. 4. CV - score at each iteration for $population\_size = 30$.

V. Analysis and Observations

From the evaluated result, we can see that our Grey Wolf optimizer indeed reduced the number of features without loss of performance. However, it did not gain any boost in metrics. Although, they are a bit better than for all features, it can be a matter of randomness of Decision Tree fitting algorithm. Moreover, GWO took too much time to end. So, we cannot

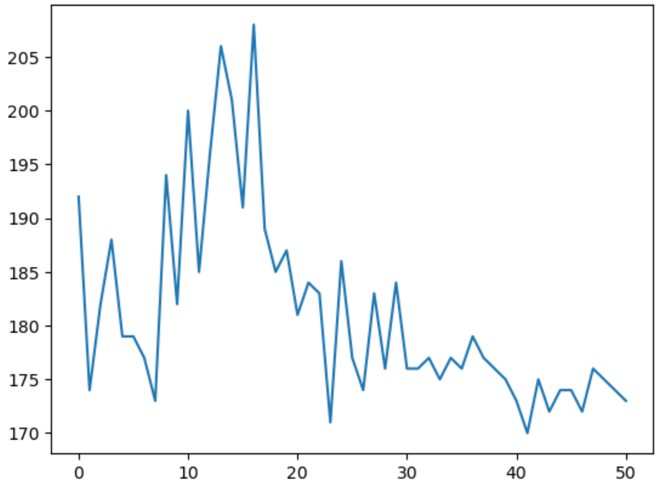

Fig. 5. Number of features chosen by GWO at each iteration for $population\_size = 10$.

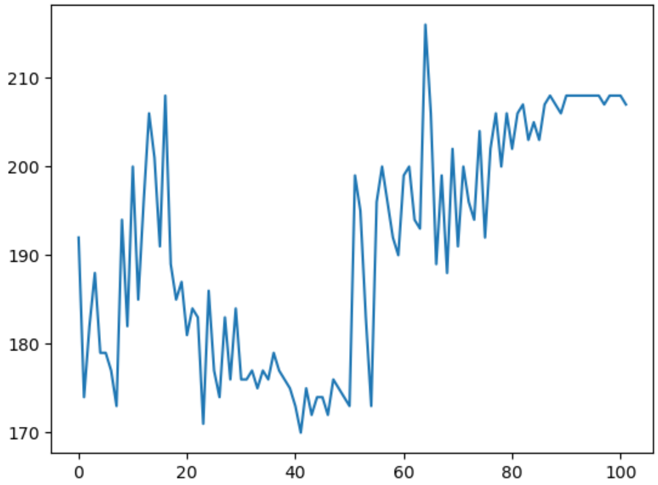

Fig. 6. Number of features chosen by GWO at each iteration for $population\_size = 30$.

TABLE I
Results

Algorithm	*Balanced accuracy*	*Precision*	*Recall*	*F1*
All features	79.42	11.8	80.74	20.59
GWO with 10 species	79.67	11.77	81.49	20.57
GWO with 30 species	79.63	11.71	81.56	20.48
RFE with 0.5 features	78.94	11.48	80.32	20.1

TABLE II
Running times of algorithms

Algorithm	*Time*
All features	No feature selection
GWO with 10 species	48 mins
GWO with 30 species	1 hr 55 mins
RFE with 0.5 features	7 mins 18 secs

consider GWO as an effective way of feature selection, but it may be for other datasets.

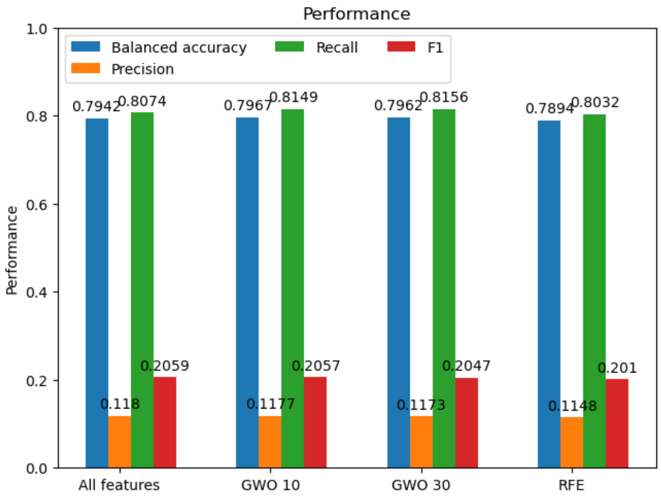

Fig. 7. Comparison of performance of different feature selection techniques.

VI. Conclusion

We implemented the Grey Wolf Optimizer nature-inspired technique as the feature selection technique and evaluated it on transaction fraud dataset with the usage of Decision tree Model and compared its performance with RFE method and a model trained on the all features. GWO did not give a significant boost in accuracy, precision,recall, or F1 score. However, it did reduced the number of features without any loss of performance but at cost of very big running time. We can conclude that GWO is not useful for feature selection in transaction fraud detection. It may perform better for other datasets. Though, we reccommend to use other feature selection techniques that are more common and faster such as Recursive Feature Elimination.

VII. Source code

The source code of the project is uploaded on GitHub and can be visited at https://github.com/seoful/grey_wolf_optimization_NIC

References

[1] Venkatesh, B., & Anuradha, J. (2019). A review of feature selection and its methods. Cybernetics and Information Technologies, 19(1), 3–26. https://doi.org/10.2478/cait-2019-0001

[2] Shirgave, S., Awati, C., More, R., & Patil, S. (2019). A review on credit card fraud detection using machine learning. International Journal of Scientific & technology research, 8(10), 1217-1220.

[3] Chaudhary, K., Yadav, J., & Mallick, B. (2012). A review of fraud detection techniques: Credit card. International Journal of Computer Applications, 45(1), 39-44.

[4] Maniraj, S. P., Saini, A., Ahmed, S., & Sarkar, S. (2019). Credit card fraud detection using machine learning and data science. International Journal of Engineering Research, 8(9), 110-115.

[5] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf optimizer," Advances in Engineering Software, vol. 69, pp. 46–61, 2014.

[6] RamaKalyani, K., & UmaDevi, D. (2012). Fraud detection of credit card payment system by genetic algorithm. International Journal of Scientific & Engineering Research, 3(7), 1-6.

[7] Jain, Y., Tiwari, N., Dubey, S., & Jain, S. (2019). A comparative analysis of various credit card fraud detection techniques. Int J Recent Technol Eng, 7(5S2), 402-407.

[8] Lakshmi, S. V. S. S., & Kavilla, S. D. (2018). Machine learning for credit card fraud detection system. International Journal of Applied Engineering Research, 13(24), 16819-16824.

[9] P. Macinec and T. Zatko, "Using Nature Inspired Algorithms for Feature Selection in Transaction Fraud Detection." [Online]. Available: https://github.com/pmacinec/transactions-fraud-detection/blob/master/paper/main.pdf.