SIM Box Fraud Detection by Deep Learning System with ICA and Beta Divergence

Ryszard Szupiluk
Warsaw School of Economics
Warsaw, Poland *rszupi@sgh.waw.pl*

Mariusz Rafało
Warsaw School of Economics
Warsaw, Poland *mrafalo@sgh.waw.pl*

Abstract

We present a system for detecting fraud related to illegal transmission of telecommunications traffic of voice calls. This phenomenon, called SIM box, can be identified and limited by using Data Mining customer classification models. The results of these models can then be decomposed by Independent Component Analysis into latent source data from which destructive components can be identified. By identifying these components using Beta Divergence, eliminating them and performing the inverse transformation to Independent Component Analysis, we can improve prediction results. The process is organized in several layers, creating a unified Deep Learning System. We demonstrate the effectiveness of the approach in a practical experiment.

Keywords: divergence functions, ICA, SIM box, fraud detection.

1. Introduction

One of the typical frauds in the mobile telecommunications market is the illegal transfer of telecommunications traffic by making voice calls bypassing the so-called interconnection rates between operators. This is performed primarily by using Fixed-Cellular Terminal (FCT) gates. The problem is called SIM box, FCT transfer, or is included in the more general category of telecommunication frauds [9]. One of the typical actions related to this problem is the identification of SIM cards on which SIM box is performed, in order to block them. The process can be performed based on the telecommunications operator's own data, using classification methods and Data Mining/Machine Learning approach. It allows to quickly and effectively create many classification/prediction models with acceptable quality. We can then transform the set of original classification results into improved form. The use of multiple model results and their transformations in a unified system is the basic idea of the proposed concept.

In this study, we propose a multi-layer system, based on Independent Component Analysis (ICA) and Beta Divergence for multi model classification improvement. The purpose of the system is to separate and eliminate interference contained in the classification results obtained with different models. The basic idea is to identify (using ICA) latent source components from the set of classification results. The components are treated as one multidimensional variable, that can include disturbances or noise responsible for classification errors. Identification of noise components by step with Beta Divergence, then their elimination and inverse transformation should lead to improved final results.

2. SIM Box fraud

The typical course of SIM box process involves purchasing a bulk number of minutes on international telecommunications platforms, then using the Internet (IP networks), bypassing official

contact gateways, introducing this traffic to the network of a given operator. Regarding the legality of this process, legal acts and the position of telecommunications regulators on the use of FCT devices are not clear and the legal basis for treating certain activities as illegal are corporate regulations for the provision of telecommunications services[8].

FCT devices enable business entities to provide services to external customers in the field of telecommunications connections without a license and without concluding specific contracts with operators. The experience of some European Union countries that have already experienced the FCT problem, indicates that attempts to legalize the use of FCT devices in the field of intermediation on the telecommunications market can be ineffective. Therefore, operators are forced to take independent steps, e.g. in the form of special provisions in the regulations.

The operator can deactivate SIM cards in the event of detection of wholesale traffic transfer by unauthorized entities. Eliminating or limiting the above phenomenon involves identifying cards transmitting illegally from the FCT device to subsequently deactivate them. Typically, identifying the use of an FCT device can be done in two ways.

First, comparing operator billing data and calculating a route for a call. This is a method used by specialist transfer detection companies. In order to identify fraud, the telephone operator provides its technical and billing platform to an external company to conduct tests. From the point of view of the telecommunications company, this may be an expensive solution, justified in a situation where there are strong grounds to conclude that such a transfer actually occurs. This usually means a preliminary assessment of the problem based on your own analytical resources.

Second, using algorithms and profiles to identify unusual behavior of operator's SIM cards. Due to the size of the active customer base, the scale of the call volume and the unpredictable timing and volume of transferred traffic, this must be done automatically and requires appropriate IT and analytical systems.

From the point of view of Data Mining/Machine Learning methods, the SIM box task is a typical classification problem and can be solved using various methods such as neural networks or decision trees algorithms. This multitude of methods used may be indicated because a relatively high accuracy of analysis is required. Cards classified as SIM box are blocked and wrong decisions can be relatively costly for the company financially and in terms of reputation. Considering the characteristics of the SIM box problem described above, there are typical variables defined for the SIM box problem, such as exclusive or large share of outgoing voice calls, no (or few) SMS, MMS and incoming calls, hidden call identification, the same device International Mobile Equipment Identity (IMEI) number, for many SIM cards, etc. Determining potentially valuable variables *a priori* allows us to expect more effective creation of classification/prediction models. As a result, many models with similar or acceptable classification quality can be expected. There is a natural possibility of using the results generated by various models to improve the final classification. This leads to systems or concepts for combining models. In general, it can be noted that most of the aggregation methods are based on averaging parameters or model results in order to minimize a specific error criterion [2]. An alternative solution may be the multi-layer system presented in this work, based on multidimensional decompositions of prediction results.

3. Noise elimination system

Let's assume we have a set of classifications obtained from different models. The set of classification results generated by different models is treated as one multidimensional variable. We assume that the each classification is contaminated with noise, interference or distortion that has specific "physical" causes, such as the inaccuracy or inadequacy of the variables explaining the phenomenon, the inappropriate form of the adopted model or the method of its optimization. Some of these factors can be common to multiple models. Their elimination should result in an overall improvement of classification quality. Since the basic idea is to search for disturbances

of a physical nature, the methods used in Blind Signal/Source Separation (BSS) seem to be adequate. This BSS problem assumes that some *a priori*, unknown signals have been mixed in an also unknown system [3, 7]. The aim of blind separation techniques is to reconstruct the original source signals from only mixed data. From many BSS techniques, we focus on the ICA method. The use of the ICA method in the context of SIM box is appropriate for several reasons. Firstly, the ICA method can be considered a general method of transformation, decomposition or data representation, regardless of its applications in the BSS problem. Secondly, it is capable of decomposing data regardless of whether the data have a time structure. Lastly, ICA allows for the definition of general generating models, such as a dynamic model of variable states.

Components identified using ICA are then classified as destructive or constructive and finally the inverse transformation to ICA is performed, i.e., a return to the prediction values. The elimination of such a destructive component should result in an improvement in the prediction measured by various criteria. The concept can be presented as a multi-layer deep learning system:

Layer 1. Creating classification (prediction) models. In this layer, we create and run classification models for the SIM box problem. The results of classification models $u_i(k)$, where $i = 1, ..., M$ stands for the model number and $k = 1, ..., K$ stands for the observation number or time index, are summarized in one multidimensional variable $\mathbf{u} = [u_1, ..., u_M]^T$.

Layer 2. Decomposition of classification models We decompose the variable $\mathbf{u}$ into hidden statistically independent source components $\mathbf{s} = [s_1, s_2, ..., s_N]$ using the ICA method. What we write as $\mathbf{s} = ICA(\mathbf{u})$, we usually take $N = M$.

Layer 3. Identification and elimination of noise components. We identify interfering components among the obtained source components. For this purpose, we use a system based on Beta divergence. Assuming that we have identified R constructive components and M-R after eliminating the destructive ones, we obtain a set of purified source components $\hat{\mathbf{s}} = [s_1, s_2, ...s_R, 0, ..., 0_M]$.

Layer 4. Inverse transformation to ICA. We perform the reverse transformation, obtaining data without noise components $\hat{\mathbf{u}} = ICA^{-1}(\hat{\mathbf{s}})$.

Layer 5. Generation of improved predictions The identified component is disabled and a new, improved prediction is generated.

4. Base component estimation - dynamic ICA

The main goal of ICA in our approach is to separate statistically independent components from the set of observations. There are many forms of ICA developments, the main differences being related to the assumptions about the model generating the observed data [7]. One of the most general ICA schemes is to assume a dynamic generating system, which means that a system that estimates independent components must also be dynamic [4].

In dynamic ICA approach, we assume that each prediction result $u_i(k)$ is a mixture of the latent components $s_j(k)$, $j, i = 1, ..., M$. The latent component can be constructive $s_t(k) = \hat{s}_t(k)$, associated with the predicted variable, or destructive $s_t(k) = \tilde{s}_t(k)$, associated with the inaccurate and missing data, imprecise estimation, unspecified distributions etc. Next, if we assume that $\mathbf{s}(k) = [\hat{s}_1(k), ..., \hat{s}_R(k), \tilde{s}_{R+1}(k), ..., \tilde{s}_M(k)]^T$ is vector of the latent components with R constructive components, the relation between observed prediction results and latent components for the dynamical linear mixing system can be represented by state space model as $\underline{\mathbf{x}}(k+1) = \underline{\mathbf{A}}\underline{\mathbf{x}}(k) + \underline{\mathbf{B}}\mathbf{s}(k)$, $\underline{\mathbf{u}}(k) = \underline{\mathbf{C}}\underline{\mathbf{x}}(k) + \underline{\mathbf{D}}\mathbf{s}(k)$, where matrices $\underline{\mathbf{A}}, \underline{\mathbf{B}}, \underline{\mathbf{C}}, \underline{\mathbf{D}} \in \mathbf{R}^{\mathbf{M}\times\mathbf{M}}$ represents parameters of the mixing system and $\mathbf{x}(k)$ is state vector [4]. The problem of Dynamic ICA is to find source signals $\mathbf{s}(k)$ and system parameters. To do this, we define an inverse system to the generating (mixing) form:

$$\mathbf{v}(k+1) = \mathbf{A}\mathbf{u}(k) + \mathbf{B}\mathbf{v}(k), \tag{1}$$

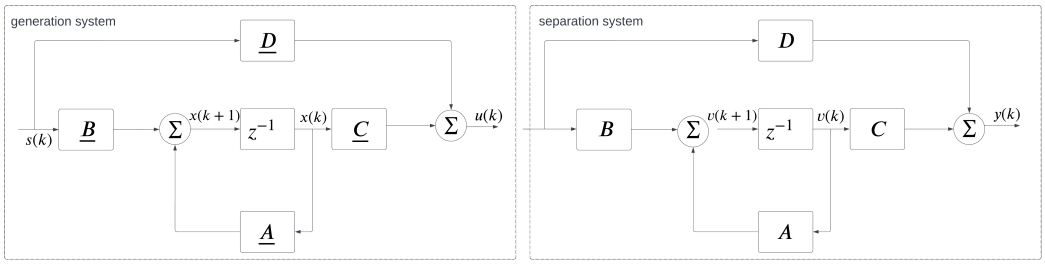

Fig. 1. Mixing and estimation systems

$$\mathbf{y}(k) = \mathbf{C}\mathbf{v}(k) + \mathbf{D}\mathbf{u}(k) \tag{2}$$

where $\mathbf{v}(k)$ is the vector of state variables of the separating system, represented by matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \in R^{M \times M}$. The matrices estimation for separation system can be performed via following rules [10]:

$$\mathbf{C}(k+1) = \mathbf{C}(k) - \eta[\phi(\mathbf{y}(k))\mathbf{v}^T(k)] \tag{3}$$

$$\mathbf{D}(k+1) = \mathbf{D}(k) + \eta[\mathbf{I} - \phi(\mathbf{y}(k))\mathbf{y}^T(k)]\mathbf{D}(k) \tag{4}$$

with

$$\phi(y_i) = -\frac{d \log p_i(y_i)}{dy_i} = -\frac{p_i'(y_i)}{p_i(y_i)}. \tag{5}$$

where where $p_i(v_i)$ denotes the probability density function of the variable y_i. The entire process, including the generating and estimating model, is presented in Figure 1.

The estimation of the matrices $\mathbf{A}$ and $\mathbf{B}$ is somewhat complex task. One of the possible solutions is to make some *a priori* assumptions about their values. The other approach can utilize information backpropagation approach. To estimate the state vector the modified Kalman filtering with hidden innovations can be used [10] .

Writing down:

$$\mathbf{W}(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A})^1\mathbf{B} + \mathbf{D} \tag{6}$$

$$\mathbf{H}(z) = \underline{\mathbf{C}}(z\mathbf{I} - \underline{\mathbf{A}})^1\underline{\mathbf{B}} + \underline{\mathbf{D}} \tag{7}$$

where: z^{-1} is the unit delay operator, we get

$$\mathbf{y}(k) = \mathbf{W}(z)\mathbf{H}(z)\mathbf{s}(k) = \mathbf{P}\Lambda(z)\mathbf{s}(k) \tag{8}$$

where $\mathbf{P}$ is permutation matrix, $\mathbf{\Lambda}$ is filtration matrix. It means that separated signals can be permuted and filtered, what are typical ambiguities for dynamic multichannel bind deconvolution methods. Assuming $s(k) \approx y(k)$, after identifying the noise components and then resetting them $s_t(k) = \tilde{s}_t(k) = 0$, we obtain a vector with pure constructive components $\hat{\mathbf{s}}(k) = [\hat{s}_1(k), ..., \hat{s}_R(k), 0_{R+1}, ..., 0_M]^T$. By substituting these components into the mixing system, we obtain improved classification results $\hat{\mathbf{u}}(k)$ as

$$\underline{\mathbf{x}}(k+1) = \underline{\mathbf{A}}\underline{\mathbf{x}}(k) + \underline{\mathbf{B}}\hat{\mathbf{s}}(k), \tag{9}$$

$$\hat{\mathbf{u}}(k) = \underline{\mathbf{C}}\underline{\mathbf{x}}(k) + \underline{\mathbf{D}}\hat{\mathbf{s}}(k). \tag{10}$$

In particular case, with null matrices $\underline{\mathbf{A}}, \underline{\mathbf{B}}, \underline{\mathbf{C}}$ separation process is reduced to standard ICA method and the filtering process is determined by the separation $\underline{\mathbf{D}} = \mathbf{D}^{-1}$ and decomposition for improved classification is

$$\hat{\mathbf{u}}(k) = \underline{\mathbf{D}}\hat{\mathbf{s}}(k). \tag{11}$$

The dynamic ICA system considered above indicates a variety of possible ways of mixing source components. In practice, however, we are most often limited to effectively operating algorithms defined for the static model (11). This is particularly appropriate for problems where the data does not have a time structure, such as the sim boxing problem.

5. Noise detection with Beta Divergence

The core aspect of the methodology involves evaluating components derived from ICA, specifically distinguishing between useful components and noise. One straightforward and potentially effective method is to test the impact of eliminating all possible subsets of these components on prediction accuracy. However, this approach encounters computational challenges, particularly for large number of components.

Given the vast number of possible model configurations, employing a some *a priori* criterion to determine whether a component is noisy or destructive may be a more feasible approach. In contexts where the data exhibits a temporal structure, such as in financial time series, it can often be the primary approach, with many solutions like autocorrelation function analysis.

In scenarios lacking temporal data structure, like customer churn, risk assessment, or fraud detection, spatial analysis becomes crucial. Experience with business problem modeling indicates that the variables involved typically do not follow distributions associated with purely random processes, such as Gaussian, uniform, or Cauchy distributions, which are known to maximize entropy under specific constraints (like constant variance or mean). Therefore, we hypothesize that a disruptive component is likely one whose distribution closely resembles these high-entropy distributions. To identify such components, we propose using a measure based on Beta divergence functions, which also considers similarities to these random distributions. This approach aims to refine the identification of noise and enhance the predictive accuracy of the models. Basic concept of Beta Divergence was defined as [1]:

$$D^{\beta}(\mathbf{y}\|\mathbf{z}) = \sum_{i=1}^{N} y_i \left(\frac{y_i^{\beta} - z_i^{\beta}}{\beta} - \frac{y_i^{\beta-1} - z_i^{\beta-1}}{\beta+1} \right) \tag{12}$$

where $\beta > 0$.

In our concept, we use its non-symmetric feature for various distributions. Comparing with the Gaussian, uniform and Cauchy distributions considered as benchmarks for destructive components, we can expect symmetric values if the components are close to or equal to these distributions $D^{\beta}(\mathbf{x}||\mathbf{y}) = D^{\beta}(\mathbf{y}||\mathbf{x})$ and asymmetric results $D^{\beta}(\mathbf{x}||\mathbf{y}) \neq D^{\beta}(\mathbf{y}||\mathbf{x})$ with different distributions. each component corresponds to a point in the space of three noise models. The noise components should be closer to zero (in terms of Euclidean distance) than the constructive ones. Achieving $D^{\beta}(\mathbf{x}||\mathbf{y}) = 0$ is unattainable, as it is not feasible for a component to simultaneously exhibit Gaussian, uniform and Cauchy distributions. However, achieving values near zero is possible. In our research, we employ divergences that incorporate weights for each individual distribution:

$$J(\mathbf{y}) = (\left(b_G \log \frac{D^{\beta}(\mathbf{y} \parallel \mathbf{s}_G)}{D^{\beta}(\mathbf{s}_G \parallel \mathbf{y})} \right)^p + \left(b_U \log \frac{D^{\beta}(\mathbf{y} \parallel \mathbf{s}_U)}{D^{\beta}(\mathbf{s}_U \parallel \mathbf{y})} \right)^p + \left(b_C \log \frac{D^{\beta}(\mathbf{y} \parallel \mathbf{s}_C)}{D^{\beta}(\mathbf{s}_C \parallel \mathbf{y})} \right)^p)^{\frac{1}{p}} \tag{13}$$

where $b_G + b_U + b_C = 1$ and $p = 2$. In this way, we examine to what extent a given distribution is similar to individual distributions or their combinations.

6. Simulation details

The empirical part of this study includes conducting a series of simulations using trained Machine Learning models. We developed a software that automates both the training of these

models and the execution of specified simulations. The program is written in the Python programming language (version 3.9) and its source code is available in the author's public GitHub repository[1]. The program is parameterized, so that it is possible to run many simulations with different assumptions[2].

We have generated synthetic CDR dataset to model various telecommunications scenarios, including SIM box fraud, overload of base stations or activity of a telecommunication probing devices. The reason of working on synthetic data is a high sensitivity of telecommunications data and the lack of available good quality CDR repositories. The dataset is created using a generator based on statistical distributions to realistically simulate the telecommunications environment. The generator was designed to reflect typical user behaviors and fraudulent activities. For this project, the dataset was synthesized for a pool of over 2 000 000 call records (one month of activity) for over 50 000 customers, encompassing a diverse range of interactions.

Table 1. Base models quality measures (average) evaluated in the study. Source: own study

Measure	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10
AUC	0.936	0.971	0.970	0.921	0.969	0.925	0.969	0.968	0.971	0.969
Sensitivity	0.919	0.996	0.997	0.927	0.994	0.895	0.992	0.989	0.918	0.989
Specificity	0.936	0.928	0.919	0.923	0.923	0.939	0.910	0.930	0.928	0.929

The simulation is based on the several steps. The first step is designed to facilitate the automated training of various deep learning models, based. The system is configured to train deep neural network models using predefined parameters such as the number of network layers, the number of neurons in each layer, the activation function in specific layer and loss function used to optimize the model. We implemented an iterative model creation process using the Monte Carlo cross-validation method, ensuring a rigorous evaluation of model quality. We created 10 models and each model was trained 50 times, during each iteration, we monitored quality measures including the confusion matrix and threshold metrics to ensure comprehensive performance assessment.

Table 2. Selected improvement ratios for ICA and Divergence Components (sample iteration)

Gaussian	Uniform	Cauchy	β	Component	Improvement	Models affected
0.18	0.82	0.0	1	c_10	0.031	7
0.33	0.67	0.0	1	c_6	0.033	2
0.06	0.86	0.08	1	c_1	0.010	9
0.42	0.58	0.0	1	c_10	0.018	4
0.14	0.86	0.0	1	c_7	0.464	1
0.01	0.99	0.0	2	c_4	0.003	3
0.67	0.33	0.0	1	c_1	0.010	8
0.73	0.27	0.0	3	c_6	0.006	1
0.02	0.98	0.0	1	c_4	0.127	9
0.75	0.21	0.04	1	c_6	0.001	2
0.07	0.93	0.0	1	c_5	0.009	5

Table 1 presents key quality measures of these models. In the process of training and validating models, we also check other quality measures, such as precision and F1 score. Additionally,

[1]https://github.com/mrafalo/ica4simbox

[2]The program was implemented using *Keras* (version 2.4.3) library. In the simulation, we used the FastICA implementation, available in the *sklearn* library (version 1.4).

after each iteration of model training, we record individual cut-off threshold values, along with the confusion matrix measures for these thresholds.

Next, we implement an ICA algorithm to identify 10 latent components. The predictions generated by models are analyzed using the ICA algorithm and the outcomes of these analyses (i.e. latent components) are standardized. Then, we identify the components that negatively affect the quality of the classification.

In the third step we identify destructive components by using two different approaches: reverse ICA and divergence measures. In a first method we go through an iterative proces and deactivate one or more ICA components by setting their values to $\mathbf{0}$. Following the exclusion of these components, we perform a backward transformation to derive new prediction results. Then, we evaluate the quality of new predictions in comparison to the original predictions from the base models. In certain cases, the deactivation of latent components led to an improvement in the AUC measure. We consider a component whose deactivation leads to improved prediction to be destructive. If identified, this destructive component appears to influence all models uniformly. The average AUC improvement observed across all models after the removal of the destructive component ranges between 5% and 16%. Sample ROC curve results are presented in Figure 2.

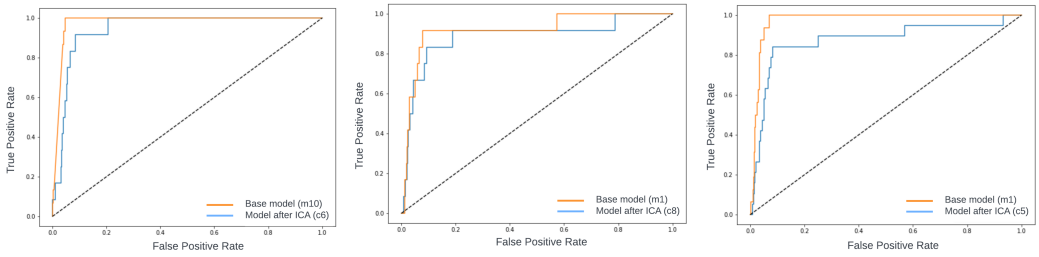

Fig. 2. Sample ROC results of backward ICA analysis. Source: own study

The second method to identifying destructive components is based on the divergence measures. These metrics are calculated over a number of iterations, each employing various parameters. We conduct tests across the two scenarios. First scenario inclludes different combinations of weights for Gaussian, uniform and Cauchy distributions; i.e. combinations of b_G, b_U and b_C in Equation 13. For each measure, we assume values from 0 (exclusion of a given distribution) to 1 (full share of the distribution), in intervals of 0.01. Second scenario includes different variants of the distance measure (i.e. different β values); we tested 5 scenarios for $\beta > 0$.

Analyses conducted on models present variability in outcomes. We evaluate the measure of weighted distance (13) across three pre-selected distributions (Gaussian, uniform and Cauchy). The behavior of the distance measure, which varies depending on the β indicator, shows slight differences as illustrated in Table 2). The table presents the beta values for which the best results were obtained and the configuration of divergence measures that was used in a given iteration. In certain iterations, notable discrepancies are observed between the distance measures for destructive and non-destructive components. The results indicate a relatively large share of the uniform distribution and the advantage of $\beta = 1$. The table also presents the average improvement value of the AUC measure and the number of models affected by a given destructive component.

Finally, for each model and for each iteration, we examined the consistency and coverage of the reverse ICA results with the divergence analysis. On average, for 40 iterations, compliance occurred in $24 - 40$ cases (which is $50\% - 75\%$). The results indicate that it is possible to identify destructive ICA components without the need to make backward predictions (which is a time-consuming and resource-intensive process). A well-parameterized Beta Divergence measure allows for much faster identification of noise components.

7. Summary

In this article, we introduce a multi-layered deep learning system for classification improvement. In the simulation, it was possible to achieve an improvement in the quality of model predictions ranging from 5% to even 16% for the selected iteration. In turn, the results for beta divergence allow for an improvement in the quality of up to 9 out of 10 models, with an improvement of 12%, up to even 46% (on average).

Due to the methodological nature of our work, we focused in the description on the layers that create novel concept. For this reason, we did not pay much attention to the first layer of creating basic classification models. They can be created based on any Machine Learning technique. However, it is important to note that any single classification model can be attached to this first layer. This means that we assume the cooperation of models, not their selection, choice or confrontation.

The choice of ICA as a method for blind separation of latent components is related to its universality and its basic role in this problem. It is possible to use other analytical techniques or, more generally, other multidimensional transformations. However, adopting other separation methods does not affect the overall approach. Due to the dominant role of ICA in the Blind Signal Separation, as well as numerous works comparing different separation methods, we did not study or compare ICA with other approaches or algorithms. This seems as essentially a separate broad research topic.

A key limitation of this study is its reliance on synthetic data, which may not fully capture the complexity of real-world scenarios. Therefore, future studies should aim to validate these findings with real-world data to enhance their applicability and reliability.

A practical outcome of this research is the demonstrated ability to improve Data Mining models by identifying and removing destructive components. This enhancement can lead to more accurate and efficient classification/prediction systems, particularly beneficial in contexts such as SIM Box fraud detection where precision is critical.

References

1. Basu, A., Harris, I.R., Hjort, N.L., Jones, M.C.: Robust and efficient estimation by minimising a density power divergence. Biometrika 85 (3), 549–559 (1998)
2. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
3. Cardoso, J.: Blind signal separation: Statistical principles. Proceedings of the IEEE 86 (10), 2009–2025 (1998)
4. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing. John Wiley, Chichester (2002)
5. Csiszar, I.,: Information measures: A critical survey. In: Transactions of the 7th Prague Conference, pp. 83–86 (1974)
6. Ighneiwa, I., Mohamed, H.S.: Bypass Fraud Detection: Artificial Intelligence Approach. (2017)
7. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley, Chichester (2001)
8. Karunathilaka, A.V.V.S.: Fraud Detection on International Direct Dial Calls. University of Colombo School of Computing, Colombo (2020)
9. Sallehuddin, R., Ibrahim, S., Mohd Zain, A., Hussein Elmi, A.: Detecting SIM Box Fraud by Using Support Vector Machine and Artificial Neural Network. Jurnal Teknologi 74, 131–143 (2015)
10. Zhang, L., Cichocki, A.: Blind Separation of Filtered Source Using State-Space Approach. Advances in Neural Information Processing Systems 11, 648–654 (1999)