

Android malware detection: An in-depth investigation of the impact of the use of imbalance datasets on the efficiency of machine learning models.

ZAKARIA SAWADOGO ^{*†}, JEAN-MARIE DEMBELE[†], GERVAIS MENDY ^{*}, SAMUEL OUYA ^{*}.

^{*}LITA(Laboratory of Computer Science, Telecommunications and Applications), Cheikh Anta Diop University of Dakar, Senegal

[†]LANI(Laboratory of Numerical Analysis and Computer Science), Gaston Berger University, Senegal
{sawadogo.zakaria,jean-marie.dembele}@ugb.edu.sn, {gervais.mendy,samuel.ouya}@ucad.edu.sn

Abstract—Machine learning techniques have become an essential part of research into the detection and classification of malicious applications. There are several approaches or algorithms that learn from existing data and predict classes. Machine learning principles recommend a balance of classes in the training dataset, but the reality on the ground is quite different. The majority of datasets used for malicious application detection are unbalanced. Class imbalance degrades classifier performance, so it is a common problem in classification tasks. This observation is much more significant in the field of Android malware detection and classification. There is little work to our knowledge on the effects of unbalanced datasets in the field of Android malware detection. Our contribution focuses on the impact of unbalanced datasets on the performance of different algorithms and the relevance of using evaluation metrics in Android malware detection. And the state of the databases from which researchers typically draw datasets. We show that for malicious application detection, some classification algorithms are not suitable for unbalanced datasets. We also prove that some of the most widely used performance evaluation metrics in the literature (Accuracy, Precision, Recall) are not very well suited to unbalanced datasets. On the other hand, the metrics (Balanced Accuracy, Geometric mean) are more suitable. These results were obtained by evaluating the performances of eleven classification algorithms as well as the adequacy of the different evaluation metrics (Accuracy, Recall, Precision, F1_score, Balanced accuracy, Matthews corcoef, Geometric mean, Fowlkes_mallows). Also not all databases are accessible by researchers and many of these databases are not updated.

Index Terms—imbalanced dataset, Android malware detection, Malware classification, Artificial intelligence, Machine learning.

I. INTRODUCTION

We are witnessing an increase in cybercrime nowadays. In response to this situation, a lot of research is being conducted in the field of cyber security, particularly in the area of detecting malicious Android applications. The introduction of artificial intelligence methods, such as machine learning and

deep learning, has significantly improved the prospects for detecting malicious Android applications. The machine learning approach requires a large amount of balanced data sets for training models. But retrieving malware for research purposes is a difficult task. The few existing databases are not updated and also have unbalanced samples. This makes the datasets used in the literature small and/or imbalanced. This could be contrary to the principle of machine learning. As stated by Cho et al. 2015 [1] a small size of the dataset affects the training of the effective model in machine learning. This is reinforced by Marceau et al. 2019 [2]. Some work has been done to correct the effects of unbalanced data sets. **Eitrich and Lang 2006**[3] have proposed an approach to efficiently optimize machine learning parameters of support vectors for unbalanced data sets. **Pozzolo et al. 2015**[4] uses Bayes minimum risk theory to find the correct classification threshold and show how to adjust it after downsampling. From imbalanced data sets, the objective functions used to learn classifiers generally tend to favor larger and less important classes. **Dong et al. 2019**[5] proposed an imbalanced class deep learning model based on a batch incremental class rectification (sparsely sampled) by extracting samples in majority classes (frequently sampled) during model training. Their model is designed to minimize the dominant effect of majority classes by uncovering sparsely sampled boundaries of minority classes in an iterative batch learning process. **Amalia et al. 2019**[6] perform an impact analysis on the performance of unbalanced datasets but focusing on the effects of data resampling. **Johnson and Khoshgoftaar 2019** [7] did a survey that highlights various gaps in deep learning from imbalanced data. They found that research in this area is very limited. We are not aware of any work analyzing the impact of imbalanced datasets on machine learning models for detecting malicious android applications. We extend the work that has been done on the impact that unbalanced datasets can have on the performance of models produced by classification algorithms[8]. Thus, this work aims to show that datasets and databases in the field of malware detection are unbalanced. Also that some classification algorithms are very sensitive to unbalanced datasets in the classification of malicious Android applications. In this analysis, we make the following contributions:

- The sensitivity status of eleven classification algorithms for the detection of android malware Applications with

Manuscript received September 21, 2022.

This work was supported by the Partnership for Applied Science, Engineering and Technology Skills - Regional Scholarship and Innovation Fund (PASET-RSIF), and is a follow-up to the invited review of the accepted and presented paper entitled "Android malware detection: Investigation of Impact of imbalanced dataset on models performance" of the 24th International Conference on Advanced Communication Technologies (ICACT2022).

Zakaria SAWADOGO, his researcher in cybersecurity including artificial at Gaston Berger University (corresponding author, email: sawadogo.zakaria@ugb.edu.sn)

Pr. Gervais MENDY is a researcher-scientist at the ESP polytechnic school at Cheikh Anta Diop University, (email: gervais.mendy@ucad.edu.sn)

Pr. Jean-Marie DEMBELE is Assimilated Professor in Computer Science at Gaston Berger University, (email: jean-marie.dembele@ugb.edu.sn)

Pr. Samuel OUYA is currently the director of the LITA laboratory at Cheikh Anta Diop University, (email: samuel.ouya@ucad.edu.sn)

imbalanced dataset.

- The analysis of the most suitable evaluation metrics for imbalanced datasets
- A review of the state of the database for android application detection.

We structured the paper as follows: After our introduction, we detailed our approach and methodology for literature selection in Section II. In Section III, we present the state of the art with a focus on the size of the datasets and the metrics used for the classification of Android malware and also a review of databases for the detection of Android malware. In section IV we present some evaluation metrics used in machine learning. Section V allows us to analyze the impact of imbalanced datasets on the performance of machine learning models. We end the paper with a conclusion in the last section VI.

II. LITERATURE SELECTION METHODOLOGY

We found it very important to review all the relevant literature. We believe it is important to describe our method of literature review. Our literature collection process was as follows:

(1) Based on the topic of our review, we determined the most relevant keywords on the basis we should do our research information. It is arguably that the information to be collected is focused on machine learning approaches for detecting Android malware. As the topic of our review is both machine learning and Android malware detection, we are summarizing some keyword combinations such as "Machine Learning + Android Malware Detection " and "imbalanced data-sets". Finally we are based on previous reviews on Android security, by researching our keywords.

(2) Our data sources were selected based on keyword searches. The vast majority of articles were from mainstream journals or major conferences, indicating that our work considered important relevant work. The following online repositories were visited: ACM Digital, IEEE Xplore , Springer Link, ScienceDirect GoogleScholar, ResearchGate, and Academia libraries.

(3) We further collected literature closely related to our review topic according to the bibliography listed in some articles. By referring to the analysis and summary of systematic literature review methods.

(4) We wanted to do a recent review. For this we have defined an article publication period ranging from 2016 to 2020 for research. We considered that the work of the last five (5) years could well summarize the research trend in fields, as new work still builds on the previous one.

III. RELATED WORKS

A. A review of some recent work

In this session, we present some recent work on Android malware detection through machine learning techniques, and then interpret the datasets and evaluation measures used by these papers. We first have 37 papers that follow our methodology. We eliminated all articles where the description of the

given games was not very precise. In the end, we worked on 27 articles. Finally we will review the literature on application databases where researchers provide their datasets.

TABLE I
A REVIEW OF SOME RECENT WORK ON MALWARE DETECTION BASED ON MACHINE LEARNING OR DEEP LEARNING.

Ref.	Dataset	benign _{r,atio}	Malware _{r,atio}	State	Metric
[9]	7520	39.47	60.53	**	Acc,F-sc
[10]	500	50	50	*	Acc
[11]	21760	91.91	8.09	**	Acc
[12]	2375	50.74	49.26	**	Acc
[13]	79729	50	50	**	Acc
[14]	200	50	50	*	Acc
[15]	3000	50	50	*	Acc
[16]	11,000	27.27	72.73	**	F-sc
[17]	2,506	86.59	13.41	**	Acc
[18]	4,000	50	50	*	Acc
[19]	10,770	48.51	51.49	**	Acc
[20]	129,013	95.69	4.31	**	Acc
[21]	41,260	51.53	48.47	**	Acc
[22]	1,958	49.13	50.87	**	Acc
[23]	23,000	43.48	56.52	**	Acc
[24]	1299	70.82	29.18	**	Acc
[25]	120,000	83.33	16.67	**	Acc, F-sc
[26]	28,760	58.41	41.59	**	Acc
[27]	24,531	54.63	45.37	**	Acc
[28]	17,219	14.44	85.56	**	Acc
[29]	27,891	50.81	49.19	**	F-sc
[30]	500	50	50	*	Acc
[31]	18,560	53.88	46.12	**	Acc
[32]	50,000	50.89	49.11	**	Pre
[33]	7,00	42.86	57.14	**	Acc
[34]	17,100	87.36	12.37	**	Acc

The table I presents twenty-six recent works in the field of Android malware detection. We have highlighted the size of the datasets used per class. And we retained the evaluation metrics that the authors put forward to evaluate their model. In the tableIII Acc= Accuracy , Pre= Precision , F-sc= F1_score.

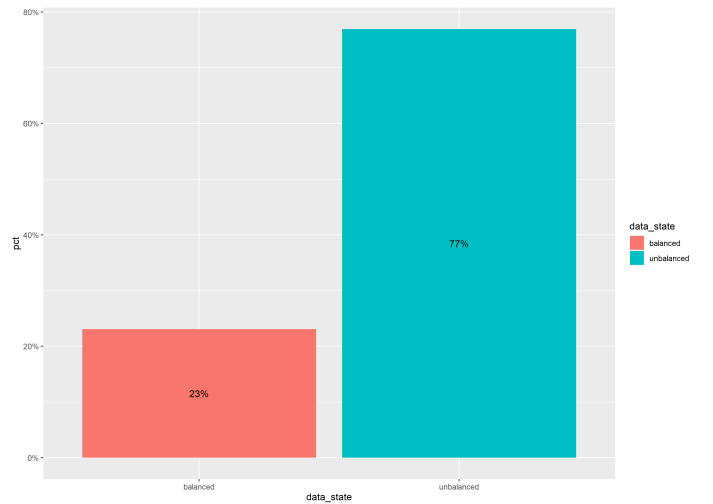


Fig. 1. The unbalanced data set rate

The figure 1 shows us that 77% of the data sets used are unbalanced and only 33% of the data sets are balanced As

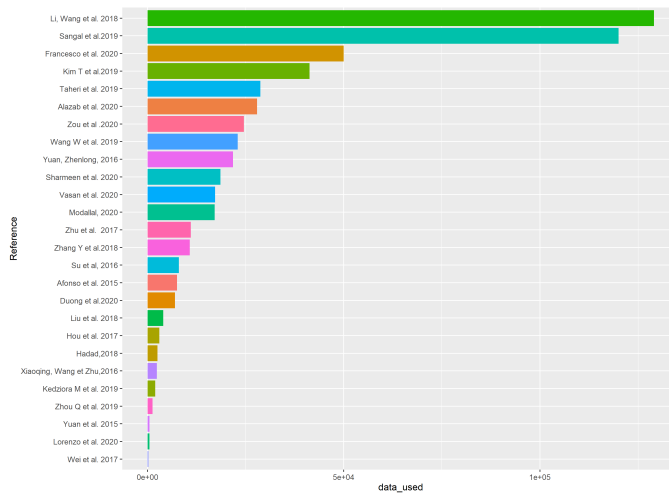


Fig. 2. The size of the data set used

shown in the figure 2 we see that the sizes of the datasets used are relatively small. Because we have a minimum dataset size which is 200 applications and 12013 applications as the maximum size.

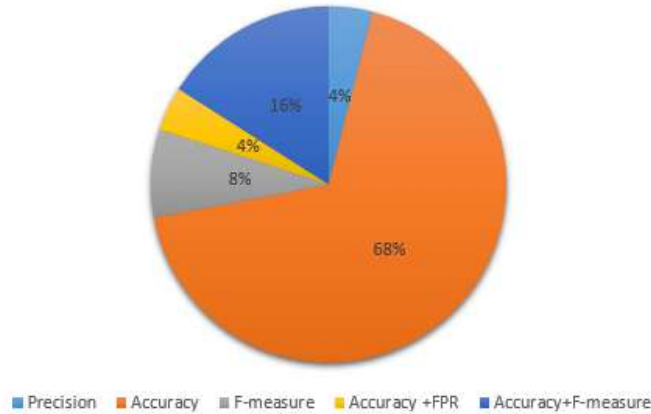


Fig. 3. The percentage of metrics used.

The figure 3 shows us the proportion of use of evaluation metrics in the recent works we considered in our analysis. Accuracy is the most used metric with 68% compared to the other metrics.

Our literature review of recent work in malicious android app detection uses unbalanced data sets (Fig.1) And these datasets are relatively small (Fig. 2). Accuracy, precision, and F1_score are the most commonly used evaluation metrics.

B. A review of some databases

The growing threat of Android malware poses a variety of security problems on the Internet. The Android malware industry is becoming increasingly troubled, with nearly thousands of new cases of Android malware every day. The availability and accessibility of databases is therefore a necessity to facilitate the work of researchers. In this

sub-section III-B, we will review the databases for android application detection. We focus on the accessibility and the updating of these databases. We also highlight the state of balance or otherwise of these databases.

The Drebin Dataset[35], this dataset contains 5,560 applications from 179 different malware families. The samples have been collected in the period of August 2010 to October 2012 and were made available to us by the MobileSandbox project.<https://www.sec.tu-bs.de/~danarp/drebin/>

Android Malware Genome Project[36], this dataset contains over 1,260 malware samples that cover the majority of existing Android malware families. Collection began in August 2010 and ended in October 2011.<http://www.malgenomeproject.org>

PRAGuard Android dataset [37], this malware dataset is made available for research purposes. The dataset contains 10479 samples, obtained by obfuscating the MalGenome and Contagio Minidump datasets with seven different obfuscation techniques <http://pralab.diee.unica.it/en/AndroidPRAGuardDataset>.

AMD Project [38], AMD contains 24,553 samples, classified into 135 varieties across 71 malware families, from 2010 to 2016. This dataset provides a picture of Android malware and is shared publicly with the community. <http://amd.arguslab.org>

CCCS-CIC-AndMal-2020 [39] The dataset includes 200,000 benign and 200,000 malware samples, for a total of 400,000 Android applications, with 14 malware categories and 191 malware families. <https://www.unb.ca/cic/datasets/andmal2020.html>

CICAndMal2017[40], is an Android malware dataset that contains both malicious and benign applications that have been run on real smartphones in order to avoid altering the execution behaviour of advanced malware samples that are able to detect the emulator environment. It was collected on over 10,854 samples (4,354 malware and 6,500 benign applications) from several[40] sources.<https://www.unb.ca/cic/datasets/andmal2017.html>

M0Droid Dataset[38], this dataset is designed to detect Android malware using signatures derived from the system call requests of individual Android APKs. The data was generated using real Android APKs as input and signatures as output.

CICMalDroid 2020[41], [42], is dataset where the samples were collected from December 2017 to December 2018 . The dataset intentionally spans five distinct categories: Adware, Banking malware, SMS malware, Riskware, and Benign. It consists of over 17,341 Android samples from several sources including the VirusTotal service, the Contagio security blog, AMD, MalDozer, and other datasets. <https://www.unb.ca/cic/datasets/maldroid-2020.html>

AndroZoo[43] is a growing collection of Android Applications collected from several sources, including the official Google Play app market. It currently contains 19,192,208 different APKs, each of which has been analysed by tens of different AntiVirus products to know which applications are detected as Malware. This dataset is provided to contribute to ongoing research efforts. <https://androzoo.uni.lu/>

In the table II, we present the status of the different databases for the detection of malicious android applications. We can observe that most of the databases are not updated. Most of them have imbalanced samples and this confirms our conclusions in the sub-section III-A.

TABLE II
A REVIEW OF SOME DATABASES

Database	Collection date	update	Samples	Accessibility
Drebin Dataset[35]	2010-2012	no update	benign :123453 Malware:5560	Currently available
Android Malware Genome Project[36]	2010-2011	no update	benign :836 Malware:1260	Currently available
Android RAGuard Dataset[37]	2015	no update	10479 samples (benign and malware)	sharing since March 2021
AMD Project[38]	2010-2016	no update	4,553 samples(benign and malware)	Currently unavailable
AndroZoo [43]	2016	Yes, regularly updated.	19,192,208 different APKs(benign and malware)	Currently available but on request
CICAnd-Mal2017[40]	2017	no update	: 6,500 malware: 4,354	Currently available
M0Droid Dataset[38]	2017	no update		Currently unavailable
CICMalDroid 2020[41]	2017-2018	no update	17,341 samples (benign and malware)	Currently available
CCCS-CIC-AndMal-2020 [39]	2020	no update	benign: 200,000 malware: 200,000	Currently available

IV. EVALUATION METRICS

For the validation of models in machine or deep learning the confusion matrix is used as evaluation criterion [44]. It is applied to classes (positive class and negative class) including the binary output (0 or 1) or multi-class. In the following, we will consider the malware class as a positive class.

The concepts of **TP**, **TN**, **FP**, **FN**, and are defined as follows.
True positive (TP): the application is a malicious application and was correctly predicted to be malicious;

True negative (TN): the application is not a malicious application and was correctly predicted to be non-malicious;

False positive (FP): the application is a malicious application but was wrongly predicted to be non-malicious.

False negative (FN): the application is not a malicious application but was wrongly predicted to be malicious;

The above four results are mutually exclusive, and thus their sum is the total number of samples in the test. Based on these four basic concepts, a series of performance metrics has been derived

A. Most common metrics

Some commonly used metrics are as follows. **Accuracy** represents the ratio of correct predictions among the total number of samples in the test. Equation 1 shows how accuracy is computed.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

Precision (P) represents the ratio of all samples correctly predicted to be positive among all samples predicted to be positive. Equation 3 shows how precision is computed.

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

Recall represents the ratio of all positive samples correctly predicted among all positive samples. Equation 4 shows how recall is computed.

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

Precision and recall are very important performance metrics, but they only provide a partial evaluation. In order to combine these two values to obtain a more complete evaluation of the performance of the classifier, the harmonic mean of precision and recall can be used, which is known as the F1 score. Equation 5 shows how the F1 score is computed

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

B. Other metrics to use with unbalanced data set

The Matthews correlation coefficient (MCC) is a more reliable statistical rate that produces a high score only if the prediction performed well in all four categories of the confusion matrix (true positives, false negatives, true negatives, and false positives) in proportion to both the size of the positives and the size of the negatives in the data set[45].

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (5)$$

Balanced accuracy is a metric used to evaluate the quality of a classifier. It is useful when the classes are unbalanced[46].

$$BA = \frac{\frac{TP}{TP + FN} + \frac{TN}{TN + FP}}{2} \quad (6)$$

Geometric Mean (GM), this metric is used to maximize the TP rate and the TN rate, while keeping the two rates relatively balanced[47].

$$GM = \sqrt{TP * TN} \quad (7)$$

The Fowlkes-Mallows(FM) index is an external evaluation method used to determine the similarity between two clusterings[48].

$$FM = \sqrt{\frac{TP}{TP + FP} \times \frac{TP}{TP + FN}} = \sqrt{Precision \times Recall} \quad (8)$$

V. IMPACT OF IMBALANCED DATASETS ON THE PERFORMANCE OF MACHINE LEARNING MODELS.

In the section III , we demonstrated that the majority of datasets used for malware detection are imbalanced. In this section, we will study the impact that datasets can have on machine learning models. To do our study, we considered eleven classification algorithms(KNN, ExtraTreeClassifier, DecisionTreeClassifier, RandomForestClassifier, AdaBoostClassifier, GradientBoosting, HistGradientBoosting, SVM, Naive Bayes, LogisticRegression, RidgeClassifier).

A. Data sets used

We used the CICMalDroid 2020 dataset[49] from several sources, including the VirusTotal service, the Contagio security blog, AMD, MalDozer and other datasets used by recent research contributions. The dataset is mainly composed of adware, bank, SMS malware, Riskware and benign. We have formed from this dataset two sub datasets((Data_A (imbalanced) and Data_B (balanced)). The table III gives the composition of the different categories.

TABLE III
DATASET MULTI-CLASS

	Data_A (imbalanced)	Data_B(balanced)
Adware	1253	1253
Banking	2100	1253
SMSMalware	3904	1253
Riskware	2546	1253
Benign	1795	1253

B. Experience environment

We conducted our experiment on an HP Z620 workstation with the following characteristics: Intel (R) Xeon (R) CPU E5-2643 0 @ 3.30 GHz 3.30GHz (2 processors), RAM: 34 GB, on which we installed Windows 10, 64 bit operating system. We used Anaconda Navigator which is a graphical user interface (GUI) which integrates several development tools such as Jupiter, Spyder and allows the use of python 3.6 and Scikit-learn, a free Python library for machine learning.

C. Implementation and analysis of results

We divided each dataset DataA, DataB in two parts. For DataA, we took 80% for the training set and 20% for the test set. We do the same thing with the DataB dataset. We trained in parallel the classification algorithms (KNN, ExtraTreeClassifier ,DecisionTreeClassifier, RandomForestClassifier, AdaBoostClassifier, GradientBoosting, HistGradientBoosting, SVM, NaiveBayes, LogisticRegression, RidgeClassifier). Then, we tested the obtained models with the test data sets.

Classifier	Accuracy	Recall	Precision	F1_score
HistGradientBoosting	95.259	94.71	94.273	94.473
RandomForest	94.353	93.542	93.079	93.274
ExtraTree	93.836	93.075	92.392	92.664
GradientBoosting	92.802	91.502	91.984	91.727
DecisionTree	89.569	87.242	87.45	87.33
AdaBoost	83.922	81.175	81.621	81.137
RidgeClassifier	75.647	70.785	78.078	73.293
KNN	68.06	60.341	61.949	60.845
Naive Bayes	56.897	48.245	61.997	49.881
LogisticRegression	55.517	49.44	52.499	46.373
SVM	48.147	36.655	59.442	35.769

Fig. 4. imbalanced Data_A

Classifier	Accuracy	Recall	Precision	F1_score
HistGradientBoosting	94.094	94.06	94.116	94.068
RandomForest	93.536	93.508	93.629	93.527
ExtraTree	92.897	92.859	92.959	92.862
GradientBoosting	92.259	92.207	92.309	92.233
DecisionTree	89.146	89.069	89.185	89.075
RidgeClassifier	70.072	69.859	73.348	70.226
AdaBoost	68.555	68.252	71.228	67.851
KNN	59.298	59.023	61.613	59.094
LogisticRegression	50.439	50.354	53.852	48.28
Naive Bayes	48.603	47.978	58.911	46.597
SVM	39.026	38.29	52.08	35.783

Fig. 5. Balanced Data_B

Figure 4 presents the results of the models obtained after training and testing with the Data_A(imbalance) data and Figure 5 with the Data_B (balance) data. We analyze these performances according to the metrics commonly used in the literature (Accuracy, Recall, precision and F1_score). We compute the percentage performance loss of models from different classification algorithms by differentiating between

the metrics obtained with unbalanced data(Data_A, fig.4) and those obtained with balanced data(Data_B, fig.5).

For the Accurcay metric, a considerable decrease in model performance for some classification algorithms. We have a loss of performance in accuracy of:

15.36% for AdaBoost ,9.12% for SVM,8.80% for KNN, 8.30% for NaiveBayes, 5.10% for RidgeClassifier, 5.60% for LogisticRegression and 1.20% for HistGradientBoosting. The other models from the algorithms (ExtraTree Classifier, Decision Tree Classifier, RandomForestClassifier, Gradient Boosting) have a loss of less than 1%.

We have a loss of performance in Precision of: 10.40% for AdaBoost, 7.30% for SVM, 4.80% for RidgeClassifier and 3.10% for NaiveBayes. The other models from the algorithms (ExtraTreeClassifier, DecisionTreeClassifier, RandomForestClassifier, GradientBoosting, KNN, LogisticRegression, HistGradientBoosting.) have a loss of less than 1%.

We have a loss of performance in Recall of: 13% for Adaboost and 1.40% for KNN . The other models from the algorithms (ExtraTreeClassifier,DecisionTreeClassifier, RandomForestClassifier, GradientBoosting , LogisticRegression, HistGradientBoosting, SVM, RidgeClassifier, NaiveBayes) have a loss of less than 1%.

We have a loss of performance in F1_score of: 13.30% for AdaBoost, 8.30% for NaiveBayes, 3.10% for RidgeClassifier. The other models from the algorithms (ExtraTreeClassifier, DecisionTreeClassifier, RandomForestClassifier, GradientBoosting, SVM, KNN, LogisticRegression, HistGradientBoosting) have a loss of less than 1%.

Classifier	Balanced_accuracy	Matthews_corrcoef	Fowlkes_mallows	Geometric mean
HistGradientBoosting	94.71	93.845	92.443	94.677
RandomForest	93.542	92.677	91.197	93.5
ExtraTree	93.075	92.021	90.414	93.021
GradientBoosting	91.502	90.631	88.338	91.431
DecisionTree	87.242	86.429	84.56	87.024
AdaBoost	81.175	79.142	76.749	80.719
RidgeClassifier	70.785	68.435	62.227	69.439
KNN	60.341	58.097	58.485	56.713
LogisticRegression	49.44	42.549	43.76	38.63
Naive Bayes	48.245	44.417	49.005	42.547
SVM	36.655	31.718	44.531	24.696

Fig. 6. imbalanced Data_A

Figure 6 presents the results of the models obtained after training and testing with the Data_A(imbalance) data and Figure 7 with the Data_B (balance) data. We analyze the performance here following the so-called balancing metrics in the literature (Balanced_accuracy, Matthews_corrcoef, Fowlkes_mallows and Geometric mean). We compute the percentage performance loss of models from different classification algorithms by differentiating between the metrics obtained with imbalanced data(Data_A, Fig.6) and those obtained with balanced data(Data_B, Fig.7).

Classifier	Balanced_accuracy	Matthews_corrcoef	Fowlkes_mallows	Geometric mean
HistGradientBoosting	94.06	92.627	88.701	94.044
RandomForest	93.508	91.938	87.621	93.49
ExtraTree	92.859	91.144	86.544	92.814
GradientBoosting	92.207	90.332	85.342	92.183
DecisionTree	89.069	86.452	79.996	88.986
RidgeClassifier	69.859	63.237	51.873	69.038
AdaBoost	68.252	61.646	53.621	65.892
KNN	59.023	49.572	41.367	57.806
LogisticRegression	50.354	39.125	36.577	45.461
Naive Bayes	47.978	38.747	37.235	42.043
SVM	38.29	28.028	34.736	30.392

Fig. 7. Balanced Data_B

We have a loss of performance in Balanced_accuracy of: 13% for AdaBoost, 1.30% for KNN. The other models from the algorithms (ExtraTreeClassifier, DecisionTreeClassifier, RandomForestClassifier, GradientBoosting, SVM, LogisticRegression, HistGradientBoosting, RidgeClassifier and NaiveBayes) have a loss of less than 1%.

We have a loss of performance in Geometric mean of: 15.82% for AdaBoost. The other models from the algorithms (ExtraTreeClassifier, DecisionTreeClassifier, RandomForestClassifier, GradientBoosting, SVM, LogisticRegression, HistGradientBoosting, KNN, RidgeClassifier and NaiveBayes) have a loss of less than 1%.

We have a performance loss with the Matthews_corrcoef metric of 0.18% to 17.5% for all considered classification algorithms.

We have a performance loss with the Fowlkes_mallows metric of 2.70% to 23.12% for all considered classification algorithms.

We have analyzed the effects of imbalanced datasets on the performance of machine learning models, particularly classification algorithms. This analysis allowed us to show that imbalanced datasets produce false performances in the models. This can be seen when comparing the evaluation metric performances obtained by the different models according to whether the dataset is imbalanced or balanced. We have differences ranging from 1% to 23% depending on the metrics considered. In view of the performance differences, we can say that some classification algorithms (KNN, AdaBoost, SVM, Naive Bayes, Logistic Regression) are not adapted to imbalanced data. Also by comparing the Balanced_accuracy, Geometric mean metrics obtained with balanced data(Fig.7) and Balanced_accuracy, Geometric mean metrics obtained with imbalanced data(Fig.6), we can say that these two metrics are more suitable for evaluating models from imbalanced data sets. In addition the Accuracy values with balanced dataset are approximate with those Balanced_accuracy obtained with imbalanced ones.

Our investigation complements other research works such as Pozzolo et al. 2015[4] ,Dong et al. 2019[5] ,Johnson and Khoshgoftaar,2019 [7], etc. which provide insights into the problem of imbalanced data.

VI. CONCLUSION

In this paper, we study the impact of unbalanced datasets on the performance of machine learning models for malicious application detection. The overall objective of our work was to show the effects of imbalanced datasets on classification algorithms. We first found through the literature that the datasets used were imbalanced. Secondly, we reviewed the android application databases for the detection of malicious applications. This showed that these databases were also imbalanced and not updated. Then we evaluated eleven classification algorithms according to the performances obtained by the commonly used metrics and by the balancing metrics. The results of this study show that some algorithms are less suitable for unbalanced datasets. Finally, we have also shown that the metrics commonly used in the literature are not always suitable. Metrics such as balanced_accuracy and Geometric mean are more suitable for imbalanced datasets.

Acknowledgment

This work was supported by The Partnership for skills in Applied Sciences, Engineering and Technology - Regional Scholarship and Innovation Fund (PASET-RSIF).

REFERENCES

- Junghwan Cho, Kyewook Lee, Ellie Shin, Garry Choy, and Synho Do. How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? 2015.
- Louis Marceau, Lingling Qiu, Nick Vandewiele, and Eric Charton. A comparison of deep learning performances with others machine learning algorithms on credit scoring unbalanced data. *arXiv*, (ML), 2019.
- Tatjana Eitrich and Bruno Lang. Efficient optimization of support vector machine learning parameters for unbalanced datasets. 196:425–436, 2006.
- AD Pozzolo et O. Caelen et RA Johnson et G. Bontempi. Calibrage de la probabilité avec sous-échantillonnage pour classification déséquilibrée. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 159–166.
- Deep learning déséquilibré par rectification incrémentielle de classe minoritaire. *Transactions IEEE sur l'analyse de modèles et l'intelligence artificielle*, 41:1367–1381.
- Amalia Luque, Alejandro Carrasco, Alejandro Martín, and Ana de las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231, 2019.
- Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1), 2019.
- Zakaria Sawadogo, Gervais Mendy, Jean Marie Dembele, and Samuel Ouya. Android malware detection: Investigating the impact of imbalanced data-sets on the performance of machine learning models. In *2022 24th International Conference on Advanced Communication Technology (ICACT)*, pages 435–441, 2022.
- Identifying Android malware using dynamically obtained features. *Journal of Computer Virology and Hacking Techniques*, 11(1):9–17, 2015.
- Zhenlong Yuan, Yongqiang Lu, Zhaoguo Wang, and Yibo Xue. Droid-Sec: Deep learning in android malware detection. *Computer Communication Review*, 44(4):371–372, 2015.
- Zhenlong Yuan, Yongqiang Lu, and Yibo Xue. Droiddetector: Android malware characterization and detection using deep learning. *Tsinghua Science and Technology*, 21(1):114–123, 2016.
- Xiaoqing Wang, Junfeng Wang, and Xiaolan Zhu. A Static Android Malware Detection Based on Actual Used Permissions Combination and API Calls. *International Journal of Computer and Information Engineering*, 10(9):1652–1659, 2016.
- Xin Su, Dafang Zhang, Wenjia Li, and Kai Zhao. A deep learning approach to android malware feature learning and detection. *Proceedings - 15th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 10th IEEE International Conference on Big Data Science and Engineering and 14th IEEE International Symposium on Parallel and Distributed Proce*, pages 244–251, 2016.
- Linfeng Wei, Weiqi Luo, Jian Weng, Yanjun Zhong, Xiaoqian Zhang, and Zheng Yan. Machine Learning-Based Malicious Application Detection of Android. 5:25591–25601, 2017.
- Shifu Hou, Aaron Saas, Lifei Chen, and Yanfang Ye. Deep4MalDroid: A deep learning framework for android malware detection based on Linux kernel system call graphs. *Proceedings - 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops, WIW 2016*, pages 104–111, 2017.
- Hui-juan Zhu, Zhu-hong You, Ze-xuan Zhu, and Wei-lei Shi. DroidDet: effective and robust detection of Android malware using static analysis along with rotation forest model. *Neurocomputing*, 2017.
- Tal Hadad, Rami Puzis, Bronislav Sidik, Nir Ofek, and Lior Rokach. *Application marketplace malware detection by user feedback analysis*, volume 867. Springer International Publishing, 2018.
- Guangyu Wang Liu and Zhijing. *Android Malware Detection Model Based on LightGBM*. 2018.
- Shaofeng Zhang and Xi Xiao. CSCdroid : Accurately Detect Android Malware via Contribution-Level-based System Call Categorization. (1), 2017.
- Dongfang Li, Zhaoguo Wang, and Yibo Xue. Fine-grained Android Malware Detection based on Deep Learning. *2018 IEEE Conference on Communications and Network Security (CNS)*, 1(L):1–2, 2018.
- Taeguen Kim, Boojoong Kang, Mina Rho, Sakir Sezer, and Eul Gyu Im. A multimodal deep learning method for android malware detection using various features. *IEEE Transactions on Information Forensics and Security*, 14(3):773–788, 2019.
- Michal Kedziora, Paulina Gawin, and Michal Szczepanik. MALWARE DETECTION USING MACHINE LEARNING ALGORITHMS AND REVERSE ENGINEERING OF ANDROID J AVA CODE. 11(1):1–14, 2019.
- Shanshan Wang, Zhenxiang Chen, Qiben Yan, Ke Ji, Lizhi Peng, Bo Yang, Mauro Conti, Shanshan Wang, Zhenxiang Chen, Qiben Yan, Ke Ji, Lizhi Peng, and Bo Yang. Information Sciences Deep and Broad URL Feature Mining for Android Malware Detection. 2019.
- Qingguo Zhou, Fang Feng, Zebang Shen, Rui Zhou, and Meng-yen Hsieh. A novel approach for mobile malware classification and detection in Android systems. (Icd), 2018.
- A L Sangal. DeepDroid : Feature Selection approach to detect Android malware using Deep Learning. pages 19–22, 2019.
- Rahim Taheri, Meysam Ghahramani, Reza Javidan, Mohammad Shojafar, Zahra Pooranian, and Mauro Conti. Similarity-based Android Malware Detection Using HammingDistance of Static Binary Features. *Future Generation Computer Systems*, 2019.
- Kwen Zou, Xi Luo, Pengfei Liu, Weiping Wang, and Haodong Wang. *ByteDroid: Android Malware Detection Using Deep Learning on Bytecode Sequences*, volume 1149 CCIS. Springer Singapore, 2020.
- Danish Vasan, Mamoun Alazab, Sobia Wassan, Hamad Naeem, Babak Safaei, and Qin Zheng. IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture. *Computer Networks*, 171(April 2019):107138, 2020.
- Moutaz Alazab, Mamoun Alazab, Andrii Shalaginov, and Abdelwaddood Mesleh. Intelligent mobile malware detection using permission requests and API calls. *Future Generation Computer Systems*, 107:509–521, 2020.
- Andrea De Lorenzo, Fabio Martinelli, Eric Medvet, Francesco Mercaldo, and Antonella Santone. Journal of Information Security and Applications Visualizing the outcome of dynamic analysis of Android malware with VizMal. *Journal of Information Security and Applications*, 50:102423, 2020.
- Shaila Sharmeen, Shamsul Huda, Jemal Abawajy, and Mohammad Mehedi. An adaptive framework against android privilege escalation threats using deep learning and semi-supervised approaches. *Applied Soft Computing Journal*, 89:106089, 2020.
- Francesco and Antonella Santone. Deep learning for image-based mobile malware detection. *Journal of Computer Virology and Hacking Techniques*, 2020.
- Detecting malicious applications on android is based on static analysis using deep learning algorithm. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(3):3494–3499, 2020.

- 34 Mohammad Modallal. Android Malware Detection Using Machine Learning with User Feedback and Static Features. 2020.
- 35 Daniel Arp, Michael Spreitzenbarth, H Malte, Hugo Gascon, and Konrad Rieck. DREBIN : Effective and Explainable Detection of Android Malware in Your Pocket.
- 36 Yajin Zhou and Xuxian Jiang. Dissecting android malware: Characterization and evolution. In *2012 IEEE Symposium on Security and Privacy*, pages 95–109, 2012.
- 37 Davide Maiorca, Davide Ariu, Igino Corona, Marco Aresu, and Giorgio Giacinto. Stealth attacks: An extended insight into the obfuscation effects on android malware. *Computers Security*, 51:16–31, 2015.
- 38 Yuping Li, Jiyong Jang, Xin Hu, and Xinming Ou. Android malware clustering through malicious payload mining. *CoRR*, abs/1707.04795, 2017.
- 39 Abir Rahali, Arash Habibi Lashkari, Gurdip Kaur, Laya Taheri, FRANCOIS GAGNON, and Frédéric Massicotte. Didroid: Android malware classification and characterization using deep image learning. In *2020 the 10th International Conference on Communication and Network Security, ICCNS 2020*, page 70–82, New York, NY, USA, 2020. Association for Computing Machinery.
- 40 Arash Habibi Lashkari, Andi Fitriah A.Kadir, Hugo Gonzalez, Kenneth Fon Mbah, and Ali A. Ghorbani. Towards a network-based framework for android malware detection and characterization. In *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, pages 233–23309, 2017.
- 41 Samaneh MahdaviFar, Andi Fitriah Abdul Kadir, Rasool Fatemi, Dima Alhadidi, and Ali A. Ghorbani. Dynamic android malware category classification using semi-supervised deep learning. In *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech)*, pages 515–522, 2020.
- 42 Zakaria Sawadogo, Gervais Mendy, Jean Marie Demebele, and Samuel Ouya. Android malware classification: Updating features through incremental learning approach(ufila). In *2022 24th International Conference on Advanced Communication Technology (ICACT)*, pages 544–550, 2022.
- 43 Kevin Allix, Tegawendé F. Bissyandé, Jacques Klein, and Yves Le Traon. Androzoo: Collecting millions of android apps for the research community. In *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*, pages 468–471, 2016.
- 44 Handling imbalanced datasets in machine learning — by baptiste — towards data science. <https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28>. Accessed: 2020-12-15.
- 45 Reza Mehmood and Arvind Selwal. *Fingerprint biometric template security schemes: Attacks and countermeasures*, volume 597. 2020.
- 46 Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PLoS ONE*, 12(6):1–17, 2017.
- 47 Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017.
- 48 E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.
- 49 Samaneh MahdaviFar, Andi Fitriah Abdul kadir, Rasool Fatemi, Dima Alhadidi, and Ali Ghorbani. Dynamic android malware category classification using semi-supervised deep learning. pages 515–522, 08 2020.



Zakaria SAWADOGO received his Master II degree in software engineering and information systems from Joseph Ki-Zerbo University in Burkina Faso. He is currently a researcher in cybersecurity including artificial intelligence and is affiliated with the Laboratoire d'Informatique, de Télécommunications et Appli-

cations (LITA) of the University Cheikh Anta Diop of Dakar and the Laboratoire d'Analyse Numérique et d'Informatique (LANI) of the University Gaston Berger of Saint Louis. His research interests include mobile security, system security and artificial intelligence. He is a member of the IEEE.



Pr. Gervais MENDY is a researcher-scientist at the ESP polytechnic school at UCAD university where he was head of the IT department from 2012 to 2016. Holder of a PhD in Computer Science from Paris-Sud XI University, his research interests are in Computer Combinatory, Social Network Analysis, Internet of Things (IoT), Artificial Intelligence and IT Security. He is a member of the LITA Laboratory of UCAD.



Pr. Jean-Marie DEMEBELE is Assimilated Professor in Computer Science at the UGB, ex Director of the UFR SAT and member of the UMI UMMISCO Senegal. He has to his credit, several scientific publications in international peer-reviewed journals. His research interests include artificial intelligence, agent-based modeling, dynamical systems, evolutionary algorithms, genetic regulatory networks, machine learning



Pr. Samuel Ouya is currently the director of the LITA laboratory at the UCAD. He was from 2013 to May 2017 the first Director of Infrastructure and Information System of the first virtual university of Senegal (UVS). Holder of a thesis in Applied Mathematics from the Gaston Berger University of Saint-Louis Senegal and a Telecommunications Thesis from the UCAD university in Dakar-Senegal, he is interested in he is interested in Applications of innovative telecom services to virtual organizations.