Comparative Analysis on Imbalanced Multi-class Classification for Malware Samples using CNN

Arwa Alzammam¹, HamadBinsalleeh¹, Basil AsSadhan², Konstantinos G. Kyriakopoulos³, SangarapillaiLambotharan³

¹Department of Computer Science, Imam Muhammad Ibn Saud Islamic University, Riyadh, Saudi Arabia
²Department of Electrical Engineering, King Saud University, Riyadh, Saudi Arabia
³Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough, UK
e-mail: asalzammam@sm.imamu.edu.sa, hmbinsalleeh@imamu.edu.sa, bssadhan@ksu.edu.sa,
K.Kyriakopoulos@lboro.ac.uk, S.Lambotharan@lboro.ac.uk

Abstract—Malware is considered as one of the main actors in cyber attacks. The number of unique malware samples is constantly on the rise; however, the ratio of benign software still greatly outnumbers malware samples. In machine learning, such datasets are known as imbalanced, where the majority class label greatly dominates over others. In this paper, we present a comparative analysis and evaluation of some of the proposed techniques in the literature in order to address the problem of classifying imbalanced multi-class malware datasets. More specifically, we use Convolutional Neural Network (CNN) as a classification algorithm to study the effect of imbalanced datasets on deep learning approaches. These experiments are conducted on three publicly available imbalanced datasets. Our performance analysis demonstrates that methods such as cost sensitive learning, oversampling and cross validation have positive effects on the model classification performance, albeit in varying degrees. Meanwhile others like using pre-trained models require more specific parameter settings. However, best practices may change in accordance with the problem domain.

Index Terms—Malware Classification, Imbalanced Dataset, Deep learning, Convolutional Neural Network

I. INTRODUCTION

A malware is a malicious software designed to intentionally perform harmful actions on computer systems. The level of damage differs based on the intention of the malware designer and the nature of the targeted systems. The number of malware samples are increasing by the day. For instance, around 350,000 new malware samples were registered every day in 2018 [1]. Despite the fast-paced proliferation of malware on a daily basis, antivirus vendors continue to be heavily dependent on signature-based detection system built by handcrafting patterns from malware codes or malware behavior in order to detect malware instances. Notably, this method confronts challenges in detecting zero-day or simple polymorphic malware employing any obfuscation mechanism.

This work was supported in part by the Gulf Science, Innovation and Knowledge Economy Programme of the U.K. Government under UK-Gulf Institutional Link Grant IL279339985

978-1-7281-4452-8/20/$31.00 ©2019 IEEE

This problem underpins the need to develop and implement automated techniques to detect malware instances. Anomaly detection mechanisms such as machine learning and deep learning help close this gap by capturing behavioral features that can be utilized in the generation of automated signatures.

Deep learning models are used to differentiate between benign and malicious behaviors, or categorize malicious behaviors into malware type or family based on the extracted features from a dataset [2]. The performance of deep learning techniques hinges on both the quality and size of the provided datasets in representing the problem to be solved. This leads to a series of challenges when applied to malware detection. The first challenge is the lack of large sized labeled dataset of malware instances [3]. The second challenge is that classification problems for malware datasets intrinsically suffer from the imbalanced dataset, where the majority class label dominates the other ones for both binary and multi-class classification. Additionally, the studies provided in the literature are primarily pertinent to imbalanced binary classification [4], [5]. Although multi-class classification problem for images using deep learning has been well covered by previous studies, most of the datasets are of balanced nature [6]. The third challenge is the limited number of systematic analysis researches on the use of different practices to resolve the overall problem of imbalance multi-class classification and in deep learning area in particular [5], [7], [8]. All these problems have prompted researchers to spend more time on specifying the most appropriate technique with a view to adopt and reduce the effects of imbalanced dataset on the classification problem. However, with the evolution of deep learning algorithms, the problem of imbalanced multi-class datasets remains one of the challenging issues [2]. In this paper, we present a comparative analysis and evaluation of some of the proposed techniques mentioned in the literature to address the problem of classifying imbalanced multi-class malware datasets. More specifically, we leverage the success of utilizing Convolutional Neural Network (CNN) classification algorithm in classifying malware samples to different malware family classes. This evaluation, in turn, is based on the two main techniques used to address the problem of imbalanced
datasets: data-level and algorithm-level [4], [7]. We conduct our experiments on three publicly available malware datasets: Malimg, Microsoft, and VirusTotal. The remainder of this paper is structured as follows: Section II provides a background on the subject of malware classification. Section III covers the datasets used in this paper. Section IV elucidates the experimental evaluation setup. Section V undertakes an assessment of the performance of the different methods used to enhance the performance for multi-class imbalanced dataset classification before discussing the results and other important key findings. Finally, conclusions and future work are provided in Section VI.

II. MALWARE CLASSIFICATION

Various neural network architectures such as Convolutional Neural Network (CNN) [9] and Recurrent Neural Network (RNN) are commonly used for malware classification [10]. For example, CNN is used to classify images of malware samples [11]. On the other hand, malware textual representation is fed to RNN to ensure better classification performance [12].

A. Convolutional Neural Network

CNN is a feed forward neural network with three layers: convolutional layer, pooling layer, and fully connected layer. The convolutional layer denotes the core layer of the CNN, where all the calculations of feature vectors and learning process are performed. Pooling layer is used to reduce the size of representations and dimensionality of data. In the fully connected layer, all outputs of the previous node are connected to produce a dimensional vector of the same size as the labelled classes [12]. Recently, malware classification problem adopted the use of CNN algorithm by feeding malware binaries as an image files, such as the works of [11], [6] and [9]. In [13], they found that malware instances belonging to the same malware family tend to share the same image structure. This observation enables the use of image representation of malware binaries to recognize variants of the same malware family. To illustrate, many contributions in the literature have used CNN for the purpose of malware classification using gray-scale images of malware binaries, such as [11] wherein they built three different convolutional neural network architectures in order to classify malware families using gray-scale images of malware binaries. In addition, they used the open-source malware dataset provided by Microsoft during the Big Data Innovation Gathering [14].

B. Imbalanced Dataset Classification

A considerable amount of time is spent on pre-processing datasets to build good machine learning models. The problem of imbalanced dataset is considered as one of the biggest issues to be resolved before undertaking any machine learning exercise. For instance, classifier models tend to be biased towards the majority class when utilizing an imbalanced dataset [4]. In [4], they categorize the techniques to minimize the affects of imbalanced datasets on the classification algorithm into two levels:

1) Data-level approaches: such as data sampling, which is considered as the most straightforward method to deal with the problem of imbalanced classification [7]. There are two main strategies to re-sample a dataset: 1) under-sampling; and 2) over-sampling. Both techniques manipulate the number of samples in each class by increasing or decreasing the samples to enable equal representation for all classes. However, the predicament with these two approaches is that they may cause the deletion of the most representative samples, which may negatively affect the performance of the classifier [4] in case of Random Under-sampling. Or, these approaches may result in an over fitting in the classification model in case of Random Over-sampling [2].

2) Algorithm-level approaches: such as cost sensitive learning, which is a method used at the classifier level to improve the performance of classifying imbalanced datasets [7]. According to [18], cost sensitive learning can be classified into two major types: 1) build cost sensitive classifiers; 2) meta-learning, which consists of two types: thresholding and sampling. Calculating class weights in loss function belongs to type “sampling”. The idea of weighting is to assign higher weights to the less represented classes, which will also have a higher mis-classification cost [19].

3) Hybrid approach: in which both data-level and algorithm-level methods are applied by using a pre-trained model, such as VGG16 and VGG19 [20]. They are also a popular approach for training a relatively small dataset, such as Malimg dataset. Pre-trained CNN models are trained on a very large image dataset, such as ImageNet [21] and are available on Keras API as applications used for feature extraction, fine tuning, or prediction [22]. Since malware datasets are small sized with training set of around 10,000 samples, multiple works have been conducted on classifying malware on pre-trained models, such as [23] and [6]. According to the work in [23], images of malware family were classified using CNN based on VGG-16 and their model was trained on two different datasets Microsoft and Malimg. Using pre-trained models for training small dataset has its share of advantages; however, they are trained on specific vision tasks and datasets. Using them on a different classification task with a different type of dataset poses a challenge. To overcome this challenge, the work in [6] modified the softmax loss to a weighted loss and then fine-tuned the model on the pre-trained VGG19. The weights are determined according to class weights in the dataset. In this paper, we examine this particular approach. To be more precise, we fine tune VGG-19 to test the positive effects acquired from using pre-trained model for the small dataset [20]. We use the pre-trained model for two tasks: 1) bottleneck features extraction and 2) fine tuning with and without class weights. Given that we are using Keras API, as opposed to modifying the loss function, we take advantage of the class weight parameter called when training the model using the “fit” method. This performs the same functionality of the weighted loss. The underlying rationale behind it is to apply weight for each class based on their representation statute within the dataset when calculating the categorical cross-entropy loss [22].

Other approaches include employing cross validation (CV)
mechanism, which is widespread in machine learning and deep learning fields. Several previous works in intrusion detection systems have applied cross validation in deep learning, such as [13], [24], [15], [25]. Using CV is known to provide a better performance than a baseline in any problem [26]. However, its usage leads to biases in the model, depending on the domain [26]. Concept drift and the similarities between malware families place malware classification in one of the domains that becomes positively biased when using CV, according to [24]. In this paper, we applied a CV to demonstrate its effects on the classification performance. Table I lists some of the previous works that have used the selected datasets or other datasets. We highlighted the type of classification algorithm, important configurations, evaluation metrics, and applied techniques in order to resolve the problem of imbalanced dataset.

C. Evaluation Metrics

The most frequently used metric to evaluate a classifier for malware classification problem is accuracy. However, there are other metrics in the literature that are known to provide more information for the evaluation. For instance, the accuracy measure suffers from some limitations in evaluating the performance for classifying imbalanced multi-class datasets. In the following, we briefly describe the used metrics:

1) **Accuracy:** measures the classifier’s ability to recognize the correct class, which is defined in terms of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), as follows:

\[
Accuracy = \frac{TP + TN}{TP + TN + FP + FN}
\]  

2) **Precision:** measures the classifier’s ability to not to classify negative samples as positive:

\[
Precision = \frac{TP}{TP + FP}
\]

3) **Recall:** measures the classifier’s ability to classify positive samples positively:

\[
Recall = \frac{TP}{TP + FN}
\]

4) **F1 score:** is considered as the harmonic mean between precision and recall:

\[
F1\_score = \frac{2 * precision \times recall}{precision + recall}
\]

5) **Loss and Weighted Loss:** Softmax activation function paired with categorical cross-entropy loss in Keras API [22] are used in this paper to calculate the Loss and Weighted Loss. The formula for the softmax loss is:

\[
SL = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} 1_{y(i) \in C_c} \log P_{y(i) \in C_c}
\]

where \( i \) denotes an event, \( c \) refers to the category, and \( P \) is the probability predicted by the model for \( i \) event of \( c \) category. The class weight for the Weighted Loss is calculated by using the scikit-learn method `compute_class_weight`. The calculated class weights are subsequently added to the class weights parameter under the Keras “fit” method; however, it is notable that this approach is applicable during training periods only.

III. DATASETS

We select three different publicly available malware datasets to compare their performance evaluation. The datasets vary in terms of the number of malware families, samples per family, and type of families. Next, we provide a brief description of each malware dataset. Table II illustrates the size, number of malware families, and number of samples in each family for each of the selected datasets. Importantly, all datasets are considered as imbalanced with a total number of 45 family classes. Following, is a brief description for each dataset.

A. Malimg Dataset

Malimg dataset contains gray-scale images of malware samples [13]. Several research papers have used this dataset to

---

**TABLE I:** Malware Image Classification Researches

| Work | Dataset | Classifier | Practices | Evaluation metrics provided | Features |
|------|---------|------------|-----------|-----------------------------|----------|
| [13] | Malimg | k-nearest neighbors | Cross validation | Accuracy: 98% | GIST |
| [6]  | Malimg | CNN | Weighted softmax + Pretrained model | Loss: 98.63% | VGG + CNN generated features |
| [11] | Microsoft | CNN | N.A | Accuracy: 99.76% | CNN generated features |
| [15] | IoTPO | CNN | Cross validation (hold-out) | Accuracy: 93.17% | CNN generated features |
| [16] | Malimg | CNN | N.A | Accuracy: 94% | CNN generated features |
| [17] | Malimg and Malheur | KNN, SVM | N.A | Accuracy: 98.84%, 98.88% | Wavelet transforms |

**TABLE II:** Datasets Size and Class Distributions

| Dataset | Size | Classes | Class Distributions |
|---------|------|---------|---------------------|
| Malimg  | 9233 | 24      | [122-116-2949-1591-198-200-146-177-142-381-431-213-184-123-159-136-162-158-80-128-132-408-97-800] |
| Microsoft | 10868 | 9      | [1013-2942-398-2478-1228-1541-42-751-475] |
| VirusTotal | 17205 | 12     | [1309-584-2288-1726-1142-788-2290-1795-1191-1319-2209-564] |
classify malware using CNN, such as [13], [6], [16] and [17]. The types of malware samples include the following: Worm, PWS, Trojan, Dialer, TrojanDownloader, and Rogue.

B. Microsoft Dataset

Microsoft dataset is considered to be the standard benchmark for malware classification, and has been cited in more than 50 research papers [14]. The dataset contains malware samples, with each sample containing two files: .asm and .byte [14]. In order to use the dataset for CNN, we convert the hexadecimal representation file for each malware into grayscale images based on the guidelines suggested by [13]. The types of malware samples include the following: Worm, Adware, Backdoor, Trojan, TrojanDownloader, and any kind of obfuscated malware.

C. VirusTotal Dataset

The third dataset is collected from VirusTotal [27]. VirusTotal refers to a public resource that provides access to malware samples in addition to other functionalities such as analyzing suspicious files, URLs, search for IP addresses, domains or file hashes. In order to label malware samples according to their family name, JSON file was generated for each malware sample and then processed using the AVClass malware labeling Tool [28]. The dataset is collected over a period of one month from 2018. Notably, malware images of this dataset are generated by converting the malware EXE into gray-scale images of a fixed width of size 256 belonging to the following malware types: Adware, Trojan, Virus, Ransomware and Worm.

IV. EXPERIMENTAL EVALUATION SETUP

We conduct a set of experiments to demonstrate the efficacy of data-level and algorithm-level techniques in addressing the problem of imbalanced multi-class malware datasets. In our experiments, we use CNN as a classification algorithm to classify malware instances into families. Figure 1 illustrates the CNN architecture which is considered to be the basic and commonly used in the previous literature [15] [29]. The experiments are conducted using Keras API with TensorFlow back-end [22]. We can summarize the steps of our experiments as follows:

1) Convert malware binaries into gray-scale images with the size of 32x32.
2) Perform the classification using the CNN for each imbalanced dataset without using any of the techniques to reduce the effects of imbalanced nature.
3) Apply all the techniques mentioned in Section II to the three different datasets separately.
4) Evaluate the performance of the CNN models based on the evaluation metrics in Section II

V. PERFORMANCE EVALUATION

We elaborate on the outcome of different experiments in order to demonstrate the effect of applying different practices mentioned in Section II that are used to enhance the performance of classifying imbalanced multi-class malware datasets. In Table III, we summarize the performance evaluation for eight main experiments conducted using the evaluation metrics in Section II. In the following, we discuss the evaluation details of these experiments:

1) **Baseline**: This is considered as the baseline where there are no treatments for the problem of imbalanced dataset. Notably, the same model configurations are used for the remaining experiments.
2) **Class weights**: The improvement in performance after adding class weights is evident in Microsoft dataset, with an increase in the accuracy by 1.2%, F1-score by 2.34%, and decrease in the loss by 12.42%. With regard to VirusTotal dataset, the added class weights have negative effects on the classification of the majority classes, which is evidenced by the decrease in the recall value by 2.31%. For example, in the per class recall performance
measurement for the baseline, the majority class, such as Gepys and Hematite, have 459 and 320 samples with 0.99% and 0.91% recall, respectively. However, when employing class weights, the recall values decline to 0.53% and 0.48%, respectively. This indicates that the majority classes are being correctly classified during the baseline experiment due to the large number of instances being fed to the classifier. Thus, once the class weights are applied, the majority classes without any significant features are affected by being misclassified. Eventually, the overall performance is found to be higher than the baseline primarily due to the increase in performance for the Precision by 4.86%, accuracy by 0.09%, and F-score by 1.23%.

3) Cross validation: Using this technique enhances the performance for accuracy and loss metrics as compared to the baseline. However, performing a cross validation introduces a positive bias. Since cross validation technique takes at least one sample from each family during the training process to enhance the accuracy of the classification, it is possible that the classifier may not be able to handle zero-day malware classification, as stated in [24]. For a multi-class malware classification, an enhanced accuracy is not the only desired outcome.

4) Under sampling: Due to the small sized datasets, the model over-fit during the training of this experiment for all datasets. Another factor that contributed to model’s over-fitting is the large number of epochs (=50), which is considered unnecessary for small sized datasets. As a result, it can be inferred that using under sampling as a practice on its own to tackle imbalanced multi-class for the problem of malware image classification is not beneficial.

5) Over sampling: Since the datasets are balanced and their size increased significantly after performing the oversampling, this method shows the best performance across all evaluation metrics for the three datasets. The oversampling technique used in this experiment is Random Oversampling.

6) Pre-trained model without class weights: Since the VGG19 model is trained on a different classification task, it was found to perform worse than the baseline.

7) Pre-trained model with class weights: The Microsoft and VirusTotal datasets show consistency in their performance when adding class weights. Their performance improved as compared to the previous experiment. However, it is less than the baseline performance for both datasets. With regard to Malimg dataset, only loss gets affected when adding class weights wherein it decreases in performance by 0.36%.

8) Bottleneck features: Both Microsoft and VirusTotal datasets were found to benefit from bottleneck features. However, it was found to have a negative effect on the Malimg dataset. The varying positive and negative reactions in the last three experiments indicate that the practice of using pre-trained model entails complexity (e.g. special parameter settings and a preprocessing step where other data factors are also considered). Therefore, it should be considered carefully when applied to malware classification.

| #Exp | Malimg | Microsoft | VirusTotal |
|------|--------|-----------|------------|
|      | Acc%   | Pre%      | F1%        | loss | Acc%   | Pre%      | F1%        | loss | Acc%   | Pre%      | F1%        | loss |
| 1    | 96.15  | 99.99     | 99.99      | 20.97 | 91.16  | 76.65     | 83.25      | 59.75 | 87.96  | 84.98     | 89.23      | 87.05 |
| 2    | 96.31  | 99.99     | 99.99      | 19.51 | 92.36  | 80.64     | 83.73      | 59.75 | 88.05  | 89.68     | 86.92      | 88.28 |
| 3    | 96.63  | 99.99     | 99.99      | 21.73 | 94.56  | 83.48     | 90.09      | 41.47 | 89.26  | 77.98     | 94.65      | 85.51 |
| 4    | 88.54  | 99.99     | 99.99      | 60.16 | 81.81  | 79.99     | 33.33      | 13.64 | 80.87  | 83.99     | 78.35      | 81.08 |
| 5    | 99.94  | 99.99     | 99.99      | 0.23  | 98.31  | 95.43     | 98.65      | 35.49 | 96.06  | 96.72     | 95.82      | 96.27 |
| 6    | 88.25  | 99.99     | 94.73      | 43.14 | 89.09  | 78.57     | 73.68      | 36.14 | 66.14  | 77.45     | 51.53      | 61.89 |
| 7    | 88.25  | 99.99     | 94.73      | 43.52 | 89.28  | 79.16     | 72.72      | 36.14 | 65.73  | 78.16     | 52.30      | 62.67 |
| 8    | 78.06  | 92.30     | 99.99      | 54.01 | 90.87  | 85.57     | 85.14      | 27.47 | 88.00  | 85.82     | 85.52      | 84.66 |

After conducting the experiments on the three available datasets, we can conclude that before deciding to use a specific practice to eliminate the imbalance multi-class classification problem, it is important to: **First**, understand the main purpose of training your classifier for malware classification. In general, the primary objective behind using deep learning for malware classification is to train the machine in order to recognize zero-day malware. **Second**, study your dataset characteristics along with other data factors. In this regard, [30] conducted a theoretical study on the impact of data imbalanced in relation to other data factors. They suggested that before attempting imbalance recovery methods to address the issue of imbalanced datasets, it is important to consider other factors that may possibly affect the classification performance degradation, such as noise, overlapping, and small disjuncts. For example, in our case, VirusTotal dataset is expected to have a better performance since it contains more samples per family class, in comparison to the other two datasets. However, we converted the malware binary file (exe files) into images without extracting the header, which, in turn, makes it vulnerable to adverse impact caused by data noise and overlapping. It is also said that overlapping has a greater effect on the classifier performance than an imbalance dataset [5]. **Third**, determine your dataset imbalance factor. Multiple imbalance impact factors have been covered in the literature, such as Imbalance Ratio (IR) [31], Shannon’s entropy (H) [32], as well as the two imbalance impact factors $IB^3$ and $IB^5$ on both sample and data level, as shown in [30] for imbalanced binary classification. Imbalance Degree (ID) [33] and Likelihood Ratio (LRID) [34] for imbalanced multi-class classification. After conducting an extensive study on the three aforementioned factors, the choice of the practice should be based on the size.
of dataset, type of metrics used to measure the performance, expertise, and time available.

VI. Conclusions and Future Work

In this paper, we provide a comparative analysis to resolve the problem of imbalanced multi-class classification by utilizing practices outlined in the extant literature in order to enhance the performance of classifying imbalanced dataset. Our evaluation concluded that the use of over-sampling outperforms other techniques. However, specific-domain features must be used in order to improve the performance of classifying imbalanced malware datasets. In the future, we intend to examine the problem on a domain-specific level by applying tuning-based extracted features from malware samples to observe the positive effects imposed on the classification problem. We also plan to study the imbalance factor and data factors that may cause the degradation in performance, as well as its correlation to the domain by calculating the Imbalance Degree (ID) and Likelihood Ratio (LRID).

REFERENCES

[1] AV-TEST, “The AV-TEST Security Report 2017/18,” Magdeburg, Germany, 2018, accessed 2019-05-22. [Online]. Available: https://www.av-test.org/en/about-the-institute/publications/

[2] O. E. David and N. S. Netanyahu, “Deepsign: Deep learning for automatic malware signature generation and classification,” 2015 International Joint Conference on Neural Networks (IJCNN), Jul 2015.

[3] D. Ucci, L. Aniello, and R. Baldoni, “Survey of machine learning techniques for malware analysis,” Computers & Security, vol. 81, pp. 123–147, 2019.

[4] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, “A survey on addressing high-class imbalance in big data,” Journal of Big Data, vol. 5, no. 1, p. 42, 2018.

[5] A. Ali, S. M. Shamsuddin, A. L. Ralescu et al., “Classification with class imbalance problem: a review,” Int. J. Advanc Soft Compu. Appl., vol. 7, no. 3, pp. 176–204, 2015.

[6] S. Yue, “Imbalanced malware images classification: a CNN based approach,” CoRR, vol. abs/1708.08042, 2017.

[7] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” Neural Networks, vol. 106, pp. 249–259, 2018.

[8] A. Fernández, V. López, M. Galar, M. J. Del Jesus, and F. Herrera, “Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches,” Knowledge-based systems, vol. 42, pp. 97–110, 2013.

[9] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, “Malware traffic classification using convolutional neural network for representation learning,” in 2017 International Conference on Information Networking (ICOIN). IEEE, 2017, pp. 712–717.

[10] R. Pascaru, J. W. Stokes, H. Sanossian, M. Marinescu, and A. Thomas, “Malware classification with recurrent networks,” in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, 2015, pp. 1916–1920.

[11] D. Gibert, “Convolutional neural networks for malware classification,” University Rovira i Virgili, Tarragona, Spain, 2016.

[12] L. Deng and D. Yu, “Deep learning: Methods and applications,” Foundations and Trends in Signal Processing, vol. 7, no. 3-4, pp. 197–387, 2014.

[13] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, “Malware images: visualization and automatic classification,” in 2011 International Symposium on Visualization for Cyber Security, VisSec ’11, Pittsburgh, PA, USA, July 20, 2011, 2011, p. 4.

[14] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, “Microsoft malware classification challenge,” CoRR, vol. abs/1802.10135, 2018.

[15] J. J. Su, V. D. Vasconcellos, S. Prasad, S. Daniele, Y. Feng, and K. Sakurai, “Lightweight classification of iot malware based on image recognition,” in 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), vol. 2. IEEE, 2018, pp. 664–669.

[16] J.-S. Luo and D. C.-T. Lo, “Binary malware image classification using machine learning with local binary pattern,” in 2017 IEEE International Conference on Big Data (Big Data), 2017, pp. 4664–4667.

[17] A. Makandar and A. Patrot, “Malware class recognition using image processing techniques,” in 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI). IEEE, 2017, pp. 76–80.

[18] C. X. Ling and V. S. Sheng, “Cost-sensitive learning and the class imbalance problem,” Encyclopedia of Machine Learning. Springer, pp. 231–235, 2008.

[19] K. M. Ting, “Inducing cost-sensitive trees via instance weighting,” in European Symposium on Principles of Data Mining and Knowledge Discovery. Springer, 1998, pp. 139–147.

[20] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–255.

[22] F. Chollet, “Keras Documentation,” Keras.io, 2015. [Online]. Available: https://keras.io/

[23] M. Kalash, M. Rochan, N. Mohammed, N. D. Bruce, Y. Wang, and F. Iqbal, “Malware classification with deep convolutional neural networks,” in 2015 IEEE conference on computer vision and pattern recognition. Ieee, 2015, pp. 1–5.

[24] F. Pellegrino, F. Pierazzi, R. Jorandey, J. Kinder, and L. Cavallaro, “TESSERACT: eliminating experimental bias in malware classification across space and time,” CoRR, vol. abs/1807.07838, 2018.

[25] C. Curtisging, B. Livshits, B. G. Zorn, and C. Seifert, “ZOZZLE: fast and precise in-browser javascript malware detection,” in 20th USENIX Security Symposium, San Francisco, CA, USA, August 8-12, 2011, Proceedings, 2011.

[26] C. Schaffer, “Selecting a classification method by cross-validation,” Machine Learning, vol. 13, no. 1, pp. 135–143, 1993.

[27] “Virustotal,” 2018, accessed 2019-05-22. [Online]. Available: https://www.virustotal.com.

[28] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero, “Avclass: A tool for massive malware labeling,” in International Symposium on Research in Attacks, Intrusions, and Defenses. Springer, 2016, pp. 230–253.

[29] G. E. Dahl, J. W. Stokes, L. Deng, and D. Yu, “Large-scale malware classification using random projections and neural networks,” in IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013, 2013, pp. 3422–3426.

[30] Y. Lu, Y. Cheung, and Y. Y. Tang, “Bayes imbalance impact index: A measure of class imbalanced dataset for classification problem,” CoRR, 2019.

[31] V. García, J. S. Sánchez, and R. A. Mollineda, “On the effectiveness of preprocessing methods when dealing with different levels of class imbalance,” Knowledge-Based Systems, vol. 25, no. 1, pp. 13–21, 2012.

[32] A. Rényi et al., “On measures of entropy and information,” in Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability. The Regents of the University of California, 1961.

[33] J. Ortigosa-Hernández, I. Inza, and J. A. Lozano, “Measuring the class-imbalance extent of multi-class problems,” Pattern Recognition Letters, vol. 98, pp. 32–38, 2017.

[34] R. Zhu, Z. Wang, Z. Ma, G. Wang, and J.-H. Xue, “Lrid: A new metric of multi-class imbalance degree based on likelihood-ratio test,” Pattern Recognition Letters, vol. 116, pp. 36–42, 2018.