ADVERSARIAL LABEL-POISONING ATTACKS AND DEFENSE FOR GENERAL MULTI-CLASS MODELS BASED ON SYNTHETIC REDUCED NEAREST NEIGHBOR

Pooya Tavallali†, Vahid Behzadan⋆, Azar Alizadeh†, Aditya Ranganath†, Mukesh Singhal†

† Electrical Engineering and Computer Science, University of California, Merced, CA - 95348
⋆SAIL Lab, University of New Haven, West Haven, CT - 06516

2022 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-9620-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIP46576.2022.9897807

ABSTRACT

Machine learning models are vulnerable to data poisoning attacks whose purpose is to undermine the model's integrity. However, the current literature on data poisoning attacks mainly focuses on ad hoc techniques that are generally limited to either binary classifiers or to gradient-based algorithms. To address these limitations, we propose a novel model-free label-flipping attack based on the multi-modality of the data, in which the adversary targets the clusters of classes while constrained by a label-flipping budget. The complexity of our proposed attack algorithm is linear in time over the size of the dataset. Also, the proposed attack can increase the error up to two times for the same attack budget. Second, a novel defense technique is proposed based on the Synthetic Reduced Nearest Neighbor model. The defense technique can detect and exclude flipped samples on the fly during the training procedure. Our empirical analysis demonstrates that (i) the proposed attack technique can deteriorate the accuracy of several models drastically, and (ii) under the proposed attack, the proposed defense technique significantly outperforms other conventional machine learning models in recovering the accuracy of the targeted model.

Index Terms— Synthetic Reduced Nearest Neighbor, Adversarial Attacks, Modality , Machine Learning

1. INTRODUCTION

Machine learning models are known to be vulnerable to data poisoning attacks [1] at training-time. In such attacks, the adversary intentionally manipulates the training data by perturbing, adding, or removing training samples with the goal of deteriorating the integrity of the model, thus, resulting in under-performing models with low accuracy [2].

Alternatively, this scenario can be seen as learning under noisy data [3]. Such attacks have the intention of altering the decision boundaries of targeted model, thus, threatening the integrity of the model [4]. The modification to data is done by manipulating samples' information such as features and labels, or by inserting and removing samples. Generally, it is assumed that attacks are constrained by an attack budget to account for realistic consitions of such attacks [1, 5]. Altering the training samples can be seen as modifying the modalities of the data that generated the training samples, thus, deteriorating the consistency of the trainset and the trained model. Much of the literature in this field is focused on the poisoning attacks against specific models and their robustness against ad hoc attacks [1, 6]. Authors of [4] theoretically investigated the accuracy of classifiers under an adversarial attack that can modify a specific portion of the trainset.

Label manipulation is a common attack surface for adversaries [1]. The adversarial label manipulation attack tends to perturb the minimum number of labels (constrained by an attack budget), such that the resulting error is maximized. A baseline strategy is to perturb the labels at random. The study in [7] established that random perturbation of labels can degrade the accuracy of SVM classifiers by flipping about 40% of the labels. However, this conclusion is limited to only binary classification problems in SVM models. Authors of [7] have shown that heuristics can improve the success of an adversary in degrading the performance of the SVM.

The optimization algorithm used here is based on the EM algorithm used in [8] composed of three steps. The first step is the assignment step which consists of optimizing the regularization term and assigning the optimal label to each centroid. The update step consists of optimizing the centroids. We will show that in the update step by using a proper surrogate objective function, the malicious samples are automatically excluded. The contributions are as follows:

1. A novel label poisoning attack mechanism based on SRNN which is not limited to specific machine learning models and classification tasks.
2. A defense technique comprised of a novel regularization method and pruning strategy for eliminating maliciously perturbed training samples.
3. Superior performance of the proposed attack mechanism to similar attack/noisy techniques against a variety of well-known machine learning models and classification tasks.
4. Robustness of the proposed approach by detecting up to 80% of the malicious samples in a variety of known resilient machine learning models and classification tasks.

2. RELATED WORKS

Many papers have focused on manipulations in the feature space as the mode of the attack. Other studies in the literature focus on gradient-based methods for selecting samples to poison in the context of SVM models [9, 10]. A data poisoning attack that uses labels as surface of attack can be modeled as noise in labels. Authors in [11] present a comprehensive survey of the label noise. In [12], authors distinguish three types of label noises: Noise Completely at Random (NCAR), Noise at Random (NAR) and Noise Not at Random (NNAR). In such settings, mislabeling of the samples is related to the features of the samples and the mislabeling can happen at the decision boundaries.

In general, there are several approaches to tackle the noisy label issue. In decision trees, the split criterion can improve robustness of the tree [13]. An alternative approach is to explicitly account for noise tolerance in the training process [14, 15, 16]. However, the majority of such methods rely on strong assumptions about the probability distribution of the noise, which constraints their scope of applicability. The state of the art in poisoning attacks is mostly comprised of attacks that focus either on specific machine learning models or are at most gradient based approaches. Therefore, they might not be applicable to general machine learning models, such as decision trees and forest models.

3. PROPOSED METHOD

3.1. Preliminaries

Synthetic Reduced Nearest Neighbor(SRNN): The efficacy and performance of nearest neighbor and reduced nearest neighbor methods for classification is well-established [17, 18]. A class of approaches that aim to address learning prototypes/centroids (i.e., synthetic samples) as the nearest neighbor model are [8]. Assume a dataset of samples consisting of N tuples of observation, $\{(x_i, y_i)\}_{i=1}^{N}$. $x_i \in \mathbb{R}^D$ is the i^{th} sample's features and its target response is $y_i \in \{1, 2, 3, ..., M\}$. The SRNN model consists of a set of centroids $C = \{(c_j, \hat{y_j})\}_{j=1}^{K}$. At the test time, prediction of an input is the label of closest centroid to that input. The optimization of the SRNN model using 0-1 loss is

$$\min_{\{(c_j,\hat{y_j})\}_1^K} \sum_{i=1}^{N} L(y_i, NN(x_i)) \tag{1}$$

$$\text{subject to } \begin{cases} NN(x_i) = \hat{y_{j^*}} \\ j_i^* = \arg\min\limits_{\{j\}_1^K} d(x_i - c_j) \end{cases} \tag{2}$$

where, $NN(\cdot)$ represents the nearest neighbor function. $d(\cdot)$ is a distance metric (chosen to be the Euclidean distance in this study). For simplicity, in the rest of this paper, we use r_{ij} for $d(x_i - c_j)$. j_i^* represents the index of closest centroid to i^{th} sample. L is a 0-1 loss function that outputs 0 if both of its input arguments are equal, otherwise, it outputs 1. The problem of (1) is in fact the problem of finding a set of K synthetic samples that achieve minimum error as a nearest neighbor model with K samples.

3.2. Modality-based adversarial label flipping

Based on SRNN model, we propose the modality-based (or cluster-based) perturbation of the training labels. The problem of selecting optimal samples for proposed attack technique is as follows:

$$\max_{\{I_i, y_i^p\}_1^N} \sum_{(x_i,y_i)\in S^{\text{train}}} L(y_i, NN^*(x_i))$$
$$\text{subject to } \begin{cases} NN^* = \arg\min\limits_{\{(c_j,\hat{y_j})\}_1^K} \sum\limits_{x_i \in S^p} L(y_i, NN(x_i)) \\ S^p = \{(x_i, (y_i(I_i - 1) + y_i^p(I_i)))\}, \\ \sum\limits_{i=1}^{N} I_i \leq Cost, \end{cases} \tag{3}$$

where NN^* represents the optimal model trained over the poisoned dataset S^p, y_i^p is the perturbed label and I_i is an indicator variable that is either 0 or 1 which is used for representing selected samples. The second constraint represents the poisoned dataset with perturbed labels. The third constraint shows the maximum allowed number of perturbations, given an attack budget ($Cost$).

The problem in eq. (3) consists of selecting samples and changing their labels to another class such that the error of NN^* is maximized over S^{train}. We propose an efficient greedy algorithm that approximates the solution for problem eq. (3) which satisfies the constraints. The samples are selected by fixing centroids of NN^* while continuing optimization only over $\{I_i, y_i^p, \hat{y_j}\}$, thereby, fixing the centroids. The problem consists of changing labels of samples in S^p such that the prediction labels of some of the centroids in NN^* are changed, increasing the error over S^{train}. This problem can be stated as

$$\max_{\{I_i, y_i^p\}_1^N, \{y_j\}_{j=1}^K} \sum_{j=1}^{K} \sum_{(x_i,y_i)\in S_j^{train}} L(y_i, \hat{y_j})$$
$$\text{subject to } \begin{cases} \hat{y_j} = \text{mode}(\{y_i\}_{S_j^p}) \forall j = 1...K \\ S^p = \{(x_i, (y_i(I_i - 1) + y_i^p(I_i)))\}, \\ \sum_{i=1}^{N} I_i \leq Cost \end{cases} \tag{4}$$

In eq. (4), S_j^p and S_j^{train} represent the trainset samples assigned to j^{th} centroid with perturbed and true labels, respectively. Note that the difference between (4) and (3) is that the centroids of clusters are fixed, thus, assignments are fixed. $\hat{y}_j$ represents the j^{th} centroid's optimal label. Assuming the cost

of changing j^{th} label is $Cost_j$, the problem can be simplified as follows:

$$\max_{\{I_j\}_{j=1}^K} \sum_{j=1}^{K} Cost_j I_j \tag{5}$$

$$\text{subject to} \sum_{j=1}^{K} Cost_j I_j < Cost, I_j = \{0,1\} \quad \forall j \in \{1 \dots K\}$$

Eq. (5) can be solved greedily by selecting from clusters with lower cost until the first constraint is violated. Other algorithms such as dynamic programming or other greedy approaches can also solve the problem in eq. (5). The proposed attack technique can have the capacity to increase the error over the trainset by two times the attack budget. **Computational complexity:** The proposed attack technique first trains a SRNN that takes $\mathcal{O}(NDK)$ [8]. Then the attack technique perturbs labels. The perturbation step takes $\mathcal{O}(N)$ which is embarrassingly fast. Compared to other label flipping attacks [10], this attack is computationally very cheap and feasible, yielding higher test errors.

3.3. Defense via Regularized Synthetic Reduced Nearest Neighbor (RSRNN)

We introduce a new parameter named *confidence range* ${r_j}_{j=1}^K$, as well as two new regularization terms for SRNN as the defense technique. Any sample beyond the confidence range, $r_{ij^*} > r_{j_i^*}$, is considered to be malicious. Note that j_i^* represents the index of closest centroid to sample i. First term consists of regularizing the confidence range for each centroid. Second regularization term consists of adding cost complexity function over the SRNN structure. The cost function facilitates the pruning of centroids and further recognizes the attacked modalities of the data. The optimization problem of training RSRNN is given in (6):

$$\min_{\{(c_j,\hat{y_j},r_j)\}_1^K} \sum_{i=1}^{N} L(y_i, NN(x_i)) + \lambda \sum_{j=1}^{K} r_j + \alpha \sum_{j=1}^{K} cost(S_j)$$

$$\text{subject to } NN(x_i) = \begin{cases} \hat{y_{j_i^*}} & r_{ij^*} < r_{j^*} \\ \text{Malicious} & \text{otherwise} \end{cases} \tag{6}$$

where, λ is the penalty coefficient of r_j, α is the cost complexity coefficient, and $cost(\cdot)$ represents the cost function of j^{th} centroid. S_j consists of the samples whose closest centroid from C is j^{th} centroid ($S_j = \{x_i | j = j_i^* \quad \forall i = 1, 2, ..., N\}$). To solve the optimization problem in (6), we follow the same EM algorithm as in [8] which was inspired by K-means algorithm [19].

3.3.1. Assignment Step:

This step has two parts. First part consists of assigning the train samples to their closest centroid. This is essentially calculating S_j for $j = 1...K$. Second part is finding optimal values to $\{\hat{y}_j\}_{j=1}^K$. The problem for j^{th} centroid can be written as

$$\min_{\hat{y_j}} \sum_{x_i \in S_j} L(y_i, \hat{y_j}) U(r_j - r_{ij}), \tag{7}$$

where $U(\cdot)$ is a step function and is used to impose the constraint in (6) for "Malicious" samples. The problem in (7) is that of finding the best constant predictor over the set of S_j. Its optimum is the most frequent label of samples in S_j ($\hat{y}_j^* = \text{mode}(\{y_i | \forall x_i \in S_j\})$).

3.3.2. Update step:

This step consists of optimizing each centroid while the centroid labels are kept constant. In this step, first, $\{r_j\}_{j=1}^K$ are fixed and $\{c_j\}_{j=1}^K$ are optimized, and then $\{c_j\}_{j=1}^K$ are fixed and ${r_j}_{j=1}^K$ are optimized. The problem of optimizing r_j for a centroid is

$$\min_{r_j} \sum_{x_i \in S_j} L(y_i, \hat{y}_j) U(r_j - r_{ij}) + U(r_{ij} - r_j) + \lambda r_j \tag{8}$$

From problem (8), it can be observed that objective function is piecewise-constant over r_j and objective function has a jump at every r_{ij}. This problem can be solved efficiently in $\mathcal{O}(|S_j| \log(|S_j|))$. This is done by sorting r_{ij} for every $x_i \in S_j$ and evaluating the objective function of (8) for every $r_j = r_{ij}$ through an incremental algorithm. In total, finding optimum r_j for all centroids is $\mathcal{O}(N \log(N))$. [8].

Finally, by iterating over update step and assignment step, the first two terms of the objective function in (6) decrease over the trainset until no further improvement can happen over the parameters. It is noteworthy that the surrogate objective function for update step also decreases the $\{cost(S_j)\}_{j=1}^K$ since it tends to create pure sets for each S_j.

3.3.3. Pruning step:

After optimizing the first two terms of (6), the third term needs further attention. For this step, all other parameters including assignment of samples are kept fixed. A clean validation set is used to prune centroids and samples of malicious modes. Finally, after removing the malicious samples and centroids, it is possible to either restart training using original SRNN over the cleaned data, or continue training with the remaining centroids and select the final model based on the error over the validation set.

4. EXPERIMENTAL RESULTS

In this section, experimental results of the proposed attack and defense techniques are presented and compared with similar techniques.

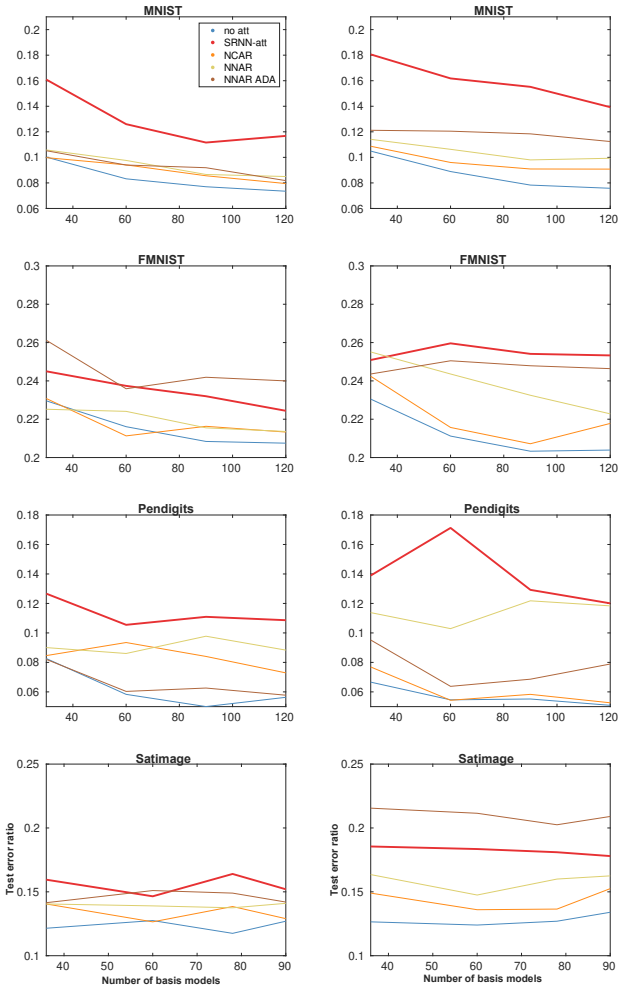

Fig. 1. Different attack techniques on different datasets. The first column shows test error for 5% attack and second column shows test error for 10% attack budget.

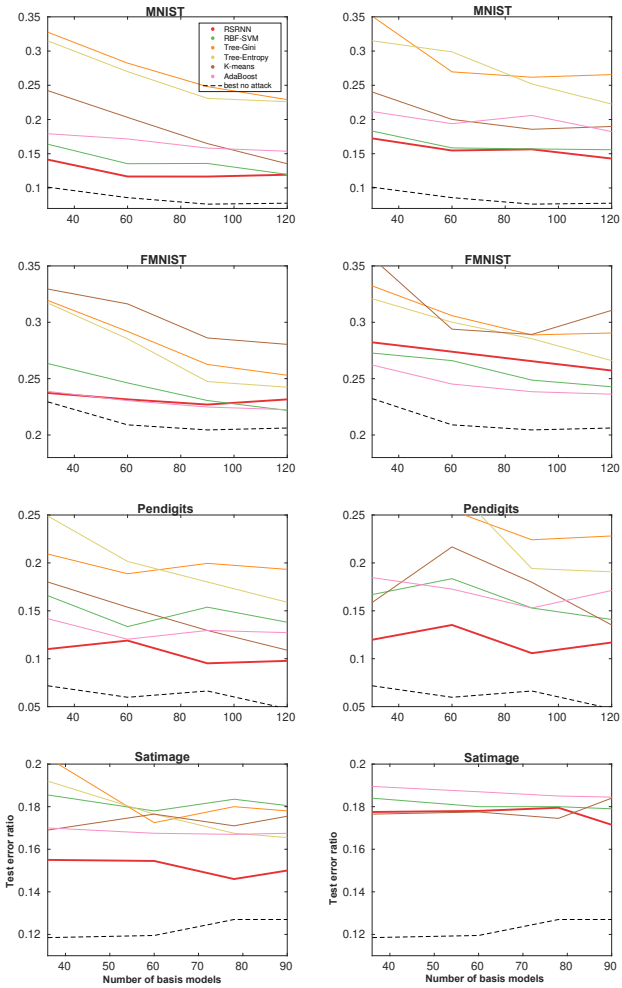

Fig. 2. Different defenses against SRNN-att model which was trained with 30 centroids. The first column and second column presents the test errors for 5% and 10% attack accuracy.

4.1. Attack Experiments

In this subsection, the performance of the modality-based adversarial label filliping (SRNN-att) is presented. The SRNN-att is compared with several other label filliping attacks that are generic and can affect all machine learning models. Accordingly, in our experiments, SRNN-att is compared with NCAR, NNAR and no attack. NCAR works by changing the labels at random. NNAR is a practical approach that aims at changing labels at the margins of decision boundaries. A state of the art model is trained and samples are predicted with low confidence by the model are targeted. In the experiments, NNAR and NNAR-ADA represent the NNAR attack using a forest model and AdaBoost model, respectively. The size of these models is selected such that they have the same size as that of the SRNN model used for SRNN-att. To evaluate the performance, SRNN, RBF-SVM, OC1, CART, K-means, AdaBoost, Nearest Neighbor, Random Nearest Neighbor are used. In order to obtain a fair comparison, the lowest error rate of each attack model among all the trained machine learning models is presented in figure 1 for each dataset in each row. In figure 1, vertical axis shows error ratio over test set and horizontal axis represents number of centroids, leaf nodes, RBFs and trees for SRNN, tree, RBF-SVM and forest models, respectively. Figure 1 shows that in all attack techniques, the SRNN-att consistently achieved the highest or comparable test error with a significant margin. Additionally, it is noteworthy that in a few instances one of the other techniques was able to increase the test error significantly but none of the other techniques was able to consistently achieve a high rise in the test error.

4.2. Defense Experiments

For the evaluation of the proposed defense against the proposed attack technique, the proposed RSRNN approach is

compared with the other state of the art models that are known to be resilient against label flipping issues. Figure 2 presents the results of experiments in this subsection for different datasets in each row. As can be observed from figure 2, RSRNN was able to constantly outperform other models with a large margin. At the same time, RSRNN was able to improve the results of SRNN by up to $2-3\%$. Additionally, RSRNN was able to detect a large portion of malicious samples up to 70% with a true positive of $50-60\%$. Finally, the size of validation set used in the experiments of this section is only 8% of the trainset.

5. CONCLUSION

In this paper, a novel data poisoning attack was proposed that is able to deteriorate the performance and undermine the integrity of state of the art machine learning models. This is the first data poisoning technique that is not limited to only binary classification, a specific model, or gradient-based approaches. The properties of this attack does not require the knowledge of the user model since its attack can affect any model. In addition, a novel defense technique based on SRNN model was proposed that is resistant against the proposed attack technique. Our experimental results showed that the RSRNN model is capable of detecting a large portion the malicious samples, thus, making it more robust to data poisoning. Finally, in experiments, RSRNN showed ability to achieve significantly lower test error compared to other known resilient models against the label flipping data poisoning.

6. REFERENCES

[1] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P Wellman, "Sok: Security and privacy in machine learning," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2018, pp. 399–414.

[2] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar, "Can machine learning be secure?," in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, 2006, pp. 16–25.

[3] Naresh Manwani and PS Sastry, "Noise tolerance under risk minimization," *IEEE transactions on cybernetics*, vol. 43, no. 3, pp. 1146–1151, 2013.

[4] Michael Kearns and Ming Li, "Learning in the presence of malicious errors," *SIAM Journal on Computing*, vol. 22, no. 4, pp. 807–837, 1993.

[5] Blaine Nelson and Anthony D Joseph, "Bounding an attack's complexity for a simple learning model," in *Proc. of the First Workshop on Tackling Computer Systems Problems with Machine Learning Techniques (SysML), Saint-Malo, France*, 2006, p. 111.

[6] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al., "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[7] Battista Biggio, Blaine Nelson, and Pavel Laskov, "Support vector machines under adversarial label noise," in *Asian conference on machine learning*, 2011, pp. 97–112.

[8] Pooya Tavallali, Peyman Tavallali, Mohammad Reza Khosravi, and Mukesh Singhal, "Interpretable synthetic reduced nearest neighbor: An expectation maximization approach," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1921–1925.

[9] Battista Biggio, Blaine Nelson, and Pavel Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[10] Shike Mei and Xiaojin Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners.," in *AAAI*, 2015, pp. 2871–2877.

[11] Benoît Frénay, Ata Kabán, et al., "A comprehensive introduction to label noise.," in *ESANN*, 2014.

[12] Benoît Frénay and Michel Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2013.

[13] Joaquín Abellán and Serafín Moral, "Building classification trees using the total uncertainty criterion," *International Journal of Intelligent Systems*, vol. 18, no. 12, pp. 1215–1225, 2003.

[14] Anil Gaba and Robert L Winkler, "Implications of errors in survey data: a bayesian model," *Management Science*, vol. 38, no. 7, pp. 913–925, 1992.

[15] Lawrence Joseph, Theresa W Gyorkos, and Louis Coupal, "Bayesian estimation of disease prevalence and the parameters of diagnostic tests in the absence of a gold standard," *American journal of epidemiology*, vol. 141, no. 3, pp. 263–272, 1995.

[16] Tim B Swartz, Yoel Haitovsky, Albert Vexler, and Tae Y Yang, "Bayesian identifiability and misclassification in multinomial data," *Canadian Journal of Statistics*, vol. 32, no. 3, pp. 285–302, 2004.

[17] Thomas Cover and Peter Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[18] Geoffrey Gates, "The reduced nearest neighbor rule (corresp.)," *IEEE transactions on information theory*, vol. 18, no. 3, pp. 431–433, 1972.

[19] Stuart Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.