

# BubbleSig: Same-Hand Ballot Stuffing Detection

Fei Zhao\*, Chengcui Zhang\*, Maya Shah†, Nitesh Saxena†

\*Department of Computer Science, The University of Alabama at Birmingham, Birmingham, USA

†Department of Computer Science and Engineering, Texas A&M University, College Station, USA

larry5@uab.edu, czhang02@uab.edu, mayashah.tam@tamu.edu, nsaxena@tamu.edu

**Abstract**—To enhance the integrity of mail-in voting, we introduce BubbleSig, an AI-assisted framework that detects same-hand ballot stuffing by analyzing “signature-like” patterns on ballot marks. This method is voter-independent that relies solely on discrepancies in marking styles across different ballots, thus preserving voter anonymity and avoiding the use of biometric or historical voter data, e.g., fingerprints or signatures. Capable of handling diverse ballot formats and layouts with a single model, BubbleSig eliminates the need for retraining across election cycles. Its efficacy is demonstrated through real election data, achieving an F1 score of 0.925 on Same-Hand Ballot Stuffing Detection dataset, 100% accuracy in both mark and ballot level stuffing detection for a small set of real ballots known filled by the same person, and notable mean Average Precision (*mAP*) and Hit Rate (*HR*) scores in retrieving ballots suspected of stuffing, using expanded test sets containing both real and synthetic ballots. Our experimental results demonstrate the model’s promise to handle diverse data collection processes, including variations in scanner types and scanning resolutions, and generalizability to various real-world ballot formats and layouts, underscoring its practical applicability. While the AI tool significantly aids in flagging potential ballot stuffing activities, final adjudications on ballot legitimacy remain with election officials, who examine suspicious ballots returned by the AI tool using the physical evidence on paper ballots.

**Index Terms**—Ballot Stuffing Detection, Deep Learning

## I. INTRODUCTION

Among various voting methods, mail-in voting with paper ballots has been popular for its simplicity and accessibility. However, this method is not without its vulnerabilities, notably the threat of same-hand ballot stuffing. This type of activity, where an individual illicitly marks multiple paper ballots, can manifest in various forms, including insertion of fraudulent/fake ballots or coercive practices, e.g., spousal coercion, where one spouse may unduly influence or **even cast a ballot on behalf of the other, often without their consent**. High-profile cases, such as Barry Morphew submitting an absentee ballot on behalf of his wife [1], underscore its direct threats to election integrity. Additionally, the 2008 Minnesota senate election [2] demonstrated how **even a single mark or a small number of them can significantly alter election results, potentially overturning the election itself**. These cases highlight the potential of same-hand ballot stuffing to compromise the accuracy of elections, emphasizing the need to strengthen existing security measures, e.g., election audits, to safeguard electoral integrity more effectively.

This work was supported by NSF CNS-2154589 and 2154507, “Collaborative Research: SaTC: CORE: Medium: Bubble Aid: Assistive AI to Improve the Robustness and Security of Reading Hand-Marked Ballots,” \$1,200,000, 10/01/2022-09/30/2026.

Our AI-assisted BubbleSig framework is designed to work as a supportive tool to existing election security measures, particularly beneficial in scenarios where historical signatures are unavailable for verification. Ballot marks can function similarly to signatures, where marks made by the same hand can exhibit distinct “signature-like” patterns. BubbleSig utilizes a Siamese-based neural network to compare marks across ballots and detects marks filled out by the same hand when it identifies similar marking styles. This pioneering solution is distinguished by its focus on analyzing ballot marks alone, a first in this domain. Its unique voter independence obviates the need for retraining the model with each new voter or election, allowing it to handle diverse elections with a single model.

The main contributions: 1. **Problem formulation and model development for detecting same-hand ballot stuffing:** we propose to formulate this ballot level detection problem as a mark/bubble level detection one, where the proposed model makes a prediction by comparing marks extracted from two different ballots directly and in a fully automatic way, without requiring any prior knowledge or biometric data from voters, e.g., fingerprints or signatures. This is also the first deep learning-based model for such purposes and has shown promising results. The detected suspicious ballots will be sent to election officials for further inspection. 2. **Voter-independent detection of ballot stuffing:** the proposed ballot stuffing detector is voter-independent. Unlike the state-of-the-art (SOTA) voter-dependent method proposed in [3], in which one model is created for each writer/voter, our method only needs one single model to handle various voters and various types of ballot layouts. There is no need to retrain the model when new voters enter the database. Furthermore, our model has shown reasonable robustness to data from different collection processes, e.g., different types of scanners and scanning resolutions, overcoming bottlenecks of traditional computer vision-based methods.

This study represents the first significant effort in detecting same-hand ballot stuffing. **Our assumption is that the attacker exhibits a passive behavior with a certain level of consistency in their marking style.** Cases where attackers deliberately alter their marking styles to evade detection fall outside our current scope but are an important avenue for future research. Despite potential marking style variations, the increased volume of fraudulently marked ballots from an attacker will inevitably lead to the emergence of discernible patterns on the attacker’s marks. These patterns enable our model to determine whether different marks are likely from

the same hand, effectively identifying stuffing activities even amidst attempts at variation. Since there is no ground truth for incidents of ballot stuffing in real-world election ballot datasets, we assume, at least during training, that two marks from the same ballots are similar (positive pairing), otherwise dissimilar (negative pairing), shown in Fig. 1. During testing, a set of real ballots marked by the same person are used to test the model. Pairings from different ballots classified as “similar” are subject to further visual inspection and verification for frauds.

## II. RELATED WORK

The SOTA methods for detecting same-hand ballot stuffing have significant limitations: 1. Manual methods verifying voter identity primarily through signature matching, are impractical for large-scale elections due to their time-consuming, labor-intensive nature, and high susceptibility to human error. 2. Biometric data-based automated methods primarily rely on biometric verification methods, such as fingerprint-matching systems. SOTA techniques, such as those in references [4] and [5], demonstrate high accuracy but raise privacy and ethical concerns due to the sensitive nature of the data required. 3. Other forms of data-based methods mainly depend on quantitative analysis of historical voting results and patterns. Those methods, e.g., [6], can identify potential electoral anomalies but often fall short in pinpointing individual fraudsters or specific instances of same-hand ballot stuffing. An example of a more targeted approach is found in [3]. The authors developed a method to identify individual bubble markers by their marking styles using a dataset of 1,840 marks from 92 participants, analyzed through PCA, Shape, and Color features using Weka’s SMO [7] classifier. This pairwise (1 vs 1) classification required  $(N^*N-1)/2$  classifiers for N classes, feasible for only small N (e.g., 92 respondents) and impractical for hundreds of thousands of voters, as each new voter necessitates retraining all  $(N^*N-1)/2$  models.

## III. METHODOLOGY

Our BubbleSig framework for detecting ballot stuffing in mail-in ballots utilizes a two-step process:

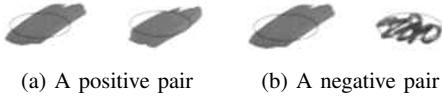


Fig. 1: Samples of positive and negative pairs

**1. Mark Extraction:** We employ the SOTA mark detection model [8] to extract marks from scanned ballots. All marks are cropped strictly according to the bounding box generated by our previous work in [8]. **2. Ballot Stuffing Detection:** The BubbleSig utilizes a Siamese-based architecture that processes pairs of marks as input and outputs a similarity score. The score ranges from 0 (not similar) to 1 (highly similar), reflecting the probability of the marks being filled out by the same hand.

The BubbleSig framework offers flexibility in choosing the encoder sub-network (backbone) with options including

VGG16, VGG19, ResNet50, ResNet101, and DensNet121. All these backbones were pre-trained on Imagenet dataset. These encoders extract features from the input mark pairs, which are then processed through different sub-networks to calculate the final similarity score.

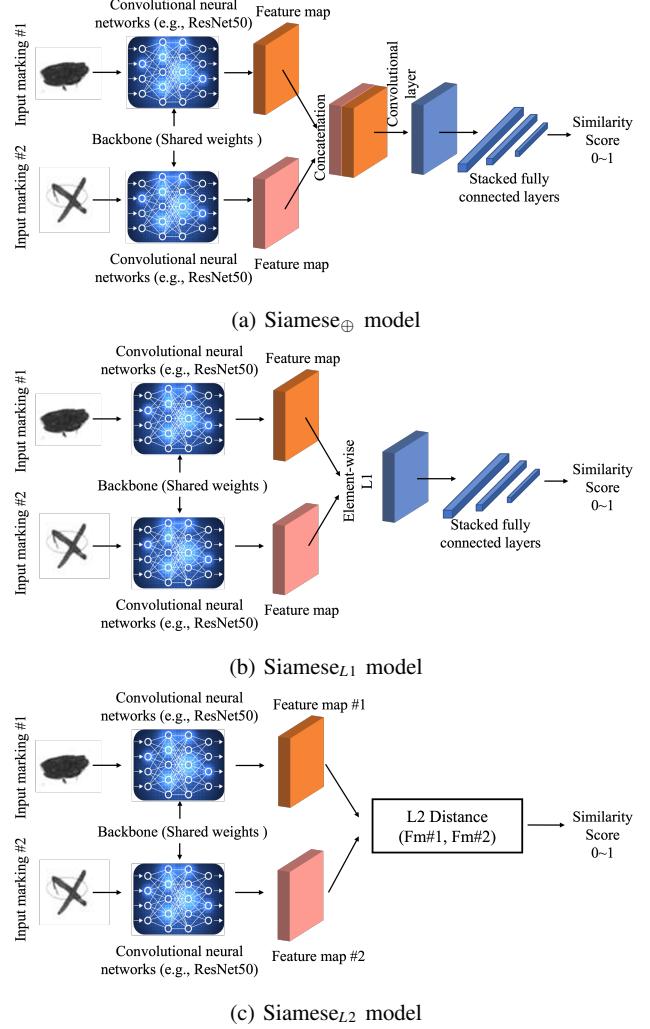


Fig. 2: The proposed ballot stuffing detection model

We present three variations of the Siamese architecture, each with its unique approach to processing the extracted features:

**Siamese $\oplus$  model (Fig. 2a):** In this variation, the high-level features from both encoders are concatenated along the channel axis. This setup allows the model to autonomously learn and leverage the relationship between the two feature maps during its training process. The concatenated features are then passed through a convolutional layer and stacked fully connected layers, culminating in a sigmoid activation function to produce the similarity score. The intuition behind this model is to enable an automated discovery of relationships between mark pairs.

**Siamese $L_1$  model (Fig. 2b):** This model computes an element-wise  $L_1$  distance between the feature maps extracted

from the two input marks. By focusing on the discrepancies between the two images, this model guides the neural network to identify subtle differences, enhancing its ability to detect ballot stuffing instance. The element-wise distance approach aims to make the model more sensitive to variations in the mark patterns. After the element-wise distance calculation, the feature maps will be flattened and sent to stacked fully connected layers to generate final predictions.

**Siamese<sub>L2</sub> model (Fig. 2c):** The third variation applies a Euclidean (L2) distance with a sigmoid activation function. This structure outputs a probability score based on the L2 distance between the feature maps, offering a direct measure of similarity. The motivation for this model is to provide a straightforward and effective way to quantify the similarity between ballot marks.

Our work uses two real-world ballot datasets: Merced County, with 7,120 RGB images (1272x2100 resolution), and Stanislaus County, with 3,151 grayscale images (1700x2800 resolution). The initial experiment was conducted on the Same-Hand Ballot Stuffing Detection dataset (BSD), created from Merced's ballots to simulate a voter-independent setting where pairs of marks from the same image are labeled positive, otherwise negative. As mentioned in Section III, the method in [8] is used to extract marks and resize them to uniform 51x51 grayscale images while preserving aspect ratios. BSD comprises 16,272 training, 2,084 validation, and 1,972 testing mark pairs, with an even split between positive and negative samples. To test our model's generalizability, we collected 464 new ballots in the Stanislaus format using two distinct setups: the first set consisted of 160 ballots filled by 5 volunteers, each contributing 32 ballots, scanned at a resolution of 5100x6600; the second set comprised 304 ballots filled by 38 volunteers, each contributing 8 ballots, scanned at 1275x1753. Different scanning resolutions and scanner types were utilized for each set to simulate diverse operational conditions. The model training was done on a single NVIDIA Tesla P100 16GB GPU with 100 epochs and an early stop.

#### IV. EXPERIMENTAL RESULTS AND DISCUSSION

For stuffing detection on the BSD dataset, as shown in Table I, the Siamese<sub>L2</sub> architecture with a DenseNet121 backbone achieved the highest F1 score of 0.925. Compared to the VGG16 and VGG19-based models, the performance of the relatively deeper networks, ResNet and DenseNet, is better. Notably, all three variations of our Siamese architecture outperformed the SOTA hand-crafted feature-based models [3]: Shape (0.673), PCA (0.602), Color (0.748), and Combined, which integrates all three features (0.744). Furthermore, the performance of the ResNet and DenseNet-based Siamese<sub>L2</sub> models was at least 21.12% better than those models. Since Siamese<sub>L1</sub> and Siamese<sub>L2</sub> outperform Siamese<sub>⊕</sub>, we use them in the subsequent study.

**1. Evaluation on real ballots filled out by same hands.** In the Stanislaus dataset never been seen during training, manual inspection found that 16 ballots (containing 46 marks) were filled out by the same election staff who manually

TABLE I: Results of ballot stuffing detection on BSD dataset

Backbone	Siamese Model Variants (F1 score)		
	Siamese <sub>⊕</sub>	Siamese <sub>L1</sub>	Siamese <sub>L2</sub>
VGG16	0.813	<b>0.859</b>	0.830
VGG19	0.850	0.864	<b>0.879</b>
ResNet50	0.872	0.885	<b>0.906</b>
ResNet101	0.874	0.896	<b>0.907</b>
DenseNet121	0.893	0.917	<b>0.925</b>

TABLE II: Mark-level ballot stuffing detection results

Model	Backbone	Accuracy/Recall	FNR
Siamese <sub>L1</sub>	ResNet50	93.744%	6.256%
	ResNet101	99.798%	0.202%
	DenseNet121	<b>100.000%</b>	<b>0.000%</b>
Siamese <sub>L2</sub>	ResNet50	99.395%	0.605%
	ResNet101	95.055%	4.9445%
	DenseNet121	99.294%	0.706%

duplicated some low-quality paper ballots, which is a common practice. We conducted three experiments on this: (a) **Mark-level detection** (Table II) tests the model on 991 mark pairs generated by the 46 marks. Each mark in a ballot forms a unique pair with each mark from every other ballot. Our model achieved at least 93% recall, validating its high sensitivity to ballot stuffing. (b) **Ballot-level detection** (Table III) evaluates the model on all the unique 120 ballot pairs generated by the 16 ballots. For each pair of ballots, each mark in one ballot is paired with each mark in the other ballot, respectively. Our model achieves a higher recall (a lower False Negative Rate, FNR) by adopting more relaxed criteria, such as at least one mark pair matching and majority vote, than requiring all mark pairs to match. However, the first two may yield a much higher FPR (False Positive Rate). (c) **Retrieval of suspected ballots** (Table IV). Each of the 16 ballots served as a query against the other 3,150 ballots in the Stanislaus dataset including the remaining 15. The similarity score for a ballot pair is aggregated from all mark pairs' scores within the ballot pair using average, median, or maximum scoring methods. We use *mAP@k* as the metric: Suppose a query ballot has *m* True matches, the system generates up to *k* recommended items/ballots ranked in decreasing order of their similarity score to the query ballot. The average precision score at *k* (*AP@k*) for this query can be calculated by:  $AP@k = \sum_{i=1}^k P(i)/\min(k, m)$  where  $P(i) = 0$  if the *i*-th ranked ballot is False and  $P(i) = Tseen_i/i$  if otherwise.  $Tseen_i$  represents the total number of True matches

TABLE III: Ballot-level ballot stuffing detection results

Criterion	Model	Accuracy/Recall	FNR
All mark pairs matching	L1	100%	0.000%
	L2	95.833%	4.167%
At least one matching	L1	100%	0.000%
	L2	100%	0.000%
Majority vote	L1	100%	0.000%
	L2	100%	0.000%

TABLE IV: The  $mAP@k$  scores on the database containing 3,151 original ballots (L1: Siamese<sub>L1</sub>, L2: Siamese<sub>L2</sub>)

Ranked by	Model	$mAP@1$	$mAP@2$	$mAP@3$	$mAP@4$	$mAP@5$	$mAP@6$	$mAP@7$	$mAP@8$	$mAP@9$
Average	L1	75.000%	70.313%	65.625%	63.672%	59.438%	<b>56.823%</b>	<b>54.190%</b>	<b>51.420%</b>	<b>47.790%</b>
	L2	62.500%	57.813%	54.514%	49.870%	43.396%	39.809%	34.632%	31.182%	27.949%
Median	L1	75.000%	68.750%	62.500%	63.672%	<b>60.688%</b>	55.087%	52.447%	49.895%	48.518%
	L2	81.250%	68.750%	63.194%	59.896%	54.917%	48.194%	45.136%	41.935%	38.511%
Maximum	L1	50.000%	57.813%	49.653%	49.349%	50.229%	48.628%	47.294%	45.777%	43.700%
	L2	<b>87.500%</b>	<b>82.813%</b>	<b>74.653%</b>	<b>64.583%</b>	56.167%	50.625%	47.985%	44.135%	40.774%

among the top  $i$  ballots. The  $mAP@k$  can be calculated as:  $mAP@k = \sum_{j=1}^N AP_j@k/N$ , in which  $N$  represents the total number of queries, and  $AP_j@k$  stands for the average precision score at  $k$  for the  $j$ -th query. The Siamese<sub>L2</sub> model (maximum score) showcased robust performance in the top-4 ranks but a drastic performance drop after rank 5. Siamese<sub>L1</sub> model (average score) demonstrates competitive performance, particularly when  $k$  value is much larger. This indicates that the average score method tends to provide a more balanced assessment across multiple mark pairs. In contrast, the maximum score method, while effective at identifying the most likely matches, might overemphasize the highest scores, leading to a sharper performance decline as the rank increases.

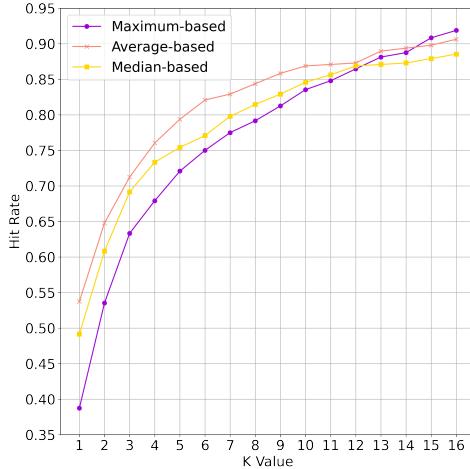


Fig. 3: Hit rate of Siamese<sub>L1</sub> model

**2. Extended evaluation with synthetic data.** As mentioned in Section III, we expanded the Stanislaus dataset to 3,615 ballots by integrating 464 ballots collected from 43 volunteers. The query set now includes 480 ballots, comprising the 464 newly collected ballots and the 16 previously existing ballots. We initially evaluated performance using  $mAP@k$ . Interestingly, this result did not mirror the optimal outcomes illustrated in Table IV, potentially due to several factors: (i) a large subset of participants (38 out of 43, or approximately 88%) contributed fewer ballots each (only 8 per volunteer), resulting in only 7 positive pairings per query out of 3,614 pairings, sharply increasing the difficulty level of the task, and (ii) a substantial decrease in image resolution in the newly added data from the second set detailed in Section III, which obscured critical details and compromised the model's effectiveness. We will explore this direction in our

future work while we continue to collect more data. In our next experiment, we use Hit Rate ( $HR$ ) as metric, which emphasizes whether the system can detect potential ballot stuffing within a reasonably sized set of top-k candidates, rather than how these candidates are ranked.  $HR$  can be formulated as:  $HR = (\sum_{i=1}^N hit_i)/N$ , where  $N$  represents the total number of queries evaluated, and  $hit_i$  is an indicator function that equals 1 if the  $i$ -th query contains at least one true-positive among the generated top-k recommendations, and 0 otherwise. As shown in Fig. 3, all methods achieve a hit rate  $\geq 72\%$  within the top 5 returned ballots, which highlights the model's adeptness at identifying stuffing activities. Notably, the Maximum-based method exhibits a sharper increase in performance as  $k$  value grows, surpassing the 90% hit rate when  $k$  reaches 15. This suggests that the Maximum-based method benefits from increased tolerance to false positives at higher  $k$  values. In contrast, the Average- and Median-based methods show more gradual improvements.

## V. CONCLUSIONS

BubbleSig, our novel AI-assisted framework utilizing Siamese architecture to process diverse ballot formats, can effectively address same-hand ballot stuffing in mail-in voting without needing retraining for new elections or voters. This pioneering research proposes a scalable, supportive tool for election officials to enhance mail-in voting integrity.

## REFERENCES

- [1] N. Vigdor, “Colorado man pleads guilty to casting missing wife’s ballot for trump.” <https://www.nytimes.com/2022/07/22/us/politics/suzanne-morphew-ballot-trump.html>, 2022. Accessed: 2024-02-07.
- [2] T. Tibbetts, “Challenged ballots: You be the judge round 1.” [https://minnesota.publicradio.org/features/2008/11/19\\_challenged\\_ballots/round1/](https://minnesota.publicradio.org/features/2008/11/19_challenged_ballots/round1/), 2008. Accessed: 2024-02-07.
- [3] J. A. Calandriño, W. Clarkson, and E. W. Felten, “Bubble trouble: Off-line de-anonymization of bubble forms.,” in *USENIX Security Symposium*, 2011.
- [4] V. Ruiz, I. Linares, A. Sanchez, and J. F. Velez, “Off-line handwritten signature verification using compositional synthetic generation of signatures and siamese neural networks,” *Neurocomputing*, vol. 374, pp. 30–41, 2020.
- [5] K. Ahrabian and B. BabaAli, “Usage of autoencoders and siamese networks for online handwritten signature verification,” *Neural Computing and Applications*, vol. 31, pp. 9321–9334, 2019.
- [6] M. Zhang, R. M. Alvarez, and I. Levin, “Election forensics: Using machine learning and synthetic data for possible election anomaly detection,” *PloS one*, vol. 14, no. 10, p. e0223950, 2019.
- [7] J. Platt, “Sequential minimal optimization: A fast algorithm for training support vector machines,” Tech. Rep. MSR-TR-98-14, Microsoft, April 1998.
- [8] F. Zhao, C. Zhang, N. Saxena, D. Wallach, and A. S. A. Rabby, “Ballot tabulation using deep learning,” in *2023 IEEE 24th International Conference on Information Reuse and Integration for Data Science (IRI)*, pp. 107–114, 2023.