Automatic Turn-Level Language Identification for Code-Switched Spanish–English Dialog

Vikram Ramanarayanan, Robert Pugh, Yao Qian & David Suendermann-Oeft

Abstract We examine the efficacy of text and speech-based features for language identification in code-switched human-human dialog interactions at the turn level. We extract a variety of character- and word-based text features and pass them into multiple learners, including conditional random fields, logistic regressors and deep neural networks. We observe that our best-performing text system significantly outperforms a majority vote baseline. We further leverage the popular i-Vector approach in extracting features from the speech signal and show that this outperforms a traditional spectral feature-based front-end as well as the majority vote baseline.

1 Introduction

Code-switching refers to multilingual speakers' alternating use of two or more languages or language varieties within the context of a single conversation or discourse in a manner consistent with the syntax and phonology of each variety [1, 2, 3, 4]. Increasing globalization and the continued rise of multilingual societies around the world makes research and development of automated tools for the processing of code-switched speech a very relevant and interesting problem for the scientific community since it has applications in multiple domains, including consumer/home electronics and business applications. In our case, an important additional motivating factor for studying and developing tools to elicit and process code-switched or crutched[1] language comes from the education domain, specifically language learning. Recent findings in the literature suggest that strategic use of code-switching of bilinguals L1 and L2 in instruction serves multiple pedagogic functions across lexical, cultural and cross-linguistic dimensions, and could enhance students' bilin-

Educational Testing Service R&D, 90 New Montgomery St, #1500, San Francisco, CA <vramanarayanan,rpugh,yqian,suendermann-oeft>@ets.org

[1] Crutching refers to language learners relying on one language to fill in gaps in vocabulary or knowledge in the other [5].

gual development and maximize their learning efficacy [6, 7]. This seems to be a particularly effective strategy especially when instructing language learners with low proficiency [8]. Therefore, the understanding of code-switched dialog and development of computational tools for automatically processing such code-switched conversations would provide an important pedagogic aid for teachers and learners in classrooms, and potentially even enhance learning at scale and personalized learning.

Automated processing of code-switched speech and dialog poses an interesting, albeit challenging problem for the scientific community. This is because the hurdles observed during traditional dialog processing tasks such as automatic speech recognition (ASR), spoken language understanding (SLU), natural language generation (NLG) and dialog management (DM) are exacerbated in the case of code-switched speech where the language the speaker is using at any given instant is not known apriori. Integrating an explicit *language identification* (or LID) step into the ASR module can alleviate these issues and improve user experience greatly. Take for example a use case of designing conversational applications for non-native English language learners (ELLs) from multiple native language (or L1) backgrounds. Many such learners tend to "crutch" on their L1 while speaking in the target language (or L2) that they are learning, especially if they are low proficiency learners [9], resulting in mixed-language speech. In such a case, LID becomes important not only for ASR, but also for DM, where the dialog designer/language expert may want the conversational agent to perform different dialog actions depending on whether the speaker used his/her L1 alone, the L2 alone, or a mixture of both during the previous turn.

Researchers have made significant progress in the automated processing of code-switched text in recent years [10, 11, 12]. Particularly relevant to our work is prior art on predicting code-switch points [13] and language identification [14, 15]. Researchers have made much progress on LID in code-switched text (tweets, in particular) thanks to recent workshops dedicated to the topic [12]. One of the top-performing systems used character n-gram, prefix and suffix features, letter case and special character features and explored logistic regression and conditional random field (CRF) learners to achieve the best performance for Spanish-English codeswitched text [16]. Yet another successful system leveraged bi-directional long short term memory networks (BLSTMs) and CRFs (along with word and character embedding features) on both Spanish-English and Standard Arabic-Egyptian language pairs [17].

While there is comparatively less work in the literature on automated analysis of code-switched speech and dialog, the number of corpora and studies is steadily growing in several language pairs – for instance, Mandarin–English [18, 19], Cantonese–English [20] and Hindi–English [21]. As far as dialog is concerned, the Bangor Corpus consists of human-human dialog conversations in Spanish–English, Welsh–English and Spanish–Welsh [22]. More recently, Ramanarayanan and Suendermann-Oeft (2017) also proposed a multimodal dialog corpus of human-machine Hindi–English and Spanish–English code-switched data [23]. To our knowledge, there is limited research on LID in code-switched speech and dialog – while certain works do use an LID system [24, 25] to improve the performance of code-mixed ASR, the LID component is baked into and uses the ASR setup. While this is perhaps the optimal way to proceed if one is only concerned with one or two language pairs, as we scale up code-switched dialog systems to multiple language pairs, building ASRs for each of the languages involved becomes difficult, especially keeping in mind SLU and DM. Hence, this paper explores an ASR-free approach to turn-level LID in code-switched dialog, exploring the efficacy of both text-based and speech-based features on a single corpus of code-switched data. To our knowledge, this is the first such exploration of both text and speech features for turn-level LID in human-human dialog data.

The rest of this paper is organized as follows: Section 2 describes the Bangor Miami corpus used for our turn-level LID experiments. We then elucidate the various text and speech features used in our experiments in Section 3, followed by the experimental setup in Section 4. Section 5 presents the results of our LID experiments as well as analyses on the different factors affecting classification accuracy. Finally, we conclude with a discussion of current observations and an outlook for future work in Section 6.

2 Data

Table 1: *Corpus statistics.*

Item	Bangor Miami
Number of turns collected	35428
Utterance-level language use or codeswitching percentage	English: 65% Spanish: 29% Both: 6%

We used the Bangor Miami corpus[2] of code-switched human–human dialog in English and Spanish for our turn-level LID experiments. The corpus consists of 56 audio recordings and their corresponding transcripts of informal conversations between two or more speakers, involving a total of 84 speakers living in Miami, Florida (USA). In total, the corpus consists of 242,475 words of text from 35 hours of recorded conversation. 63% of the transcribed words are English, 34% Spanish, and 3% are undetermined. The manual transcripts include beginning and end times of utterances and per word language identification. For our experiments, we excluded turns containing tokens with ambiguous or undetermined language.

The audio was split into turns as specified by the provided timestamps. Each turn was downsampled to 8 kHz and converted to a single channel. The transcriptions were processed by performing whitespace tokenization on each turn, and removing event descriptions (such as ”&=laugh”) and unintelligible tokens.

[2] `http://bangortalk.org.uk/`

3 Feature Extraction

We performed turn-level LID experiments using speech-only features as well as text features (which serve as a benchmark since they directly contain linguistic information), and compared them to a majority vote (or chance) baseline. In this section, we will first describe the various text and speech features explored, followed by the machine learning setup in the subsequent section.

3.1 Text Features

Following earlier work [16, 17], we experimented with the following low-level binary text features that capture the presence or absence of the following:

- **Word n-grams**: We used a bag-of-words representation, trying uni- and bigrams.
- **Character n-grams**: The set of unique character n-grams ($1 \leq n \leq 4$), without crossing word-boundaries. For example, the word sequence “la sal” would produce the following character n-grams {‘l’, ‘a’, ‘s’, ‘al’, ‘la’, ‘sa’, ‘sal’}.
- **Character Prefixes/Suffixes**: All affixes with length ≤ 3. For example, the word “intricate” would have prefixes {‘i’, ‘in’, ‘int’}, and suffixes {‘ate’, ‘te’, and ‘e’}.

Additionally, for one experiment (LSTM), we used randomly initialized word embeddings as input features, which were trained as part of the network.

3.2 Speech Features

We explored two featuresets for our speech experiments – OpenSMILE features and i-Vector features. We used the OpenSMILE toolkit [26] to extract features from the audio signal - specifically, the standard emobase2010 feature set containing 1582 features that is tuned for recognition of paralinguistic information in speech. These consist of multiple low-level descriptors - intensity, loudness, MFCCs, pitch, voicing probability, F0 envelope, Line Spectral Frequencies (LSFs) and zero crossing rate, among others - as well as their functionals (such as standard moments).

We also trained a GMM-based i-Vector system (see [27]) using the Kaldi toolkit [28]. Initially introduced for speaker recognition [29], i-Vectors have also been shown to be particularly useful features for language recognition (see for example [30]). The i-Vector extraction procedure can be viewed as a probabilistic compression process that maps input speech features into a reduced dimensionality space using a linear Gaussian model – for more details, see [29]. The front-end for the extracts were 20-dimensional MFCCs including C0, using a 20ms Hamming window with 10ms time shift along with their first and second derivatives. We deleted non-speech segments within utterances through an energy-based voice active detection (VAD) method, and performed utterance-based cepstral mean normalization on the acoustic feature vectors. We trained a GMM and a full covariance matrix as the Universal Background Model (UBM) by using the entire Fisher English [31] and Spanish corpora, in addition to the Bangor corpus data (we did not include code-switched turns). We then used the Bangor Corpus to train an i-Vector extractor T-matrix. The number of Gaussian components and the i-Vector dimensions were set to 1024 and 800, respectively.

4 Experiments

We randomly partitioned the Bangor corpus data into train and test sets using an 80%-20% split. For experiments with text-based featuresets, we first extracted the word and character level features described in Section 3. We then tried two approaches to predicting one of 3 classes – English, Spanish or Code-switched – at the turn-level: (i) Use a CRF to make word-level predictions, and aggregate them to form a turn-level prediction, and (ii) aggregate the features at the turn level and try a variety of learners, including logistic regression and deep neural networks to make language predictions at the turn level. Additionally, we tried passing sequences of word-embeddings (randomly initialized and trained on the train partition of the Bangor corpus) to an LSTM and making an LID prediction for each turn. We experimented with different learner configurations and parameter settings and summarize the best performing featureset and learner combination in the Results section.

We set up the speech experiments using the following steps:

1. Partition the full audio files into train and test sets. We hold out any code-switched turns from the training partition at this step in order to train a two-class UBM model for i-Vector extraction.
2. Segment each turn into a sequence of two-second segments, and extract an 800-dimensional i-Vector for each. For this step, 80% of the code-switched i-Vectors are randomly moved back to the training partition while the remaining 20% were moved to the test partition. (Note that during this process we ensure that we use the *same train-test partitions as we are using for the text-based systems* to enable a fair comparison of systems).
3. For each turn, we generate a three-dimensional vector consisting of a) the Euclidean distance of the segments from the average English segment, b) the distance from the average Spanish segment, and c) the length of the turn in seconds. We also experimented with using Cosine distance, as well as each segment's PLDA score for the respective classes, instead of the Euclidean distance.

4. Optionally, use SMOTE oversampling [32] to overcome class imbalance, ensuring that there Spanish and English classes have the same size (code-switched turns were not oversampled[3]).
5. Fit an appropriate learner (such as a Linear Discriminant Analysis classifier) in order to predict the turn-level language of each turn in the test set.

5 Observations and Analysis

System	Featureset	Machine Learner	F1 per class			Weighted Ave. F1
			Eng	Spa	CS	
Text	Word, Character n-grams (1-4), Character affixes (1-3) for current, previous, and next word	CRF, aggregated for turn-level predictions	0.98	0.95	0.91	**0.97**
	Word embeddings	LSTM	0.98	0.95	0.9	0.96
	Word n-grams (1-2), Character n-grams (1-4), Character affixes (1-3), turn-length in tokens	Logistic Regression	0.97	0.93	0.67	0.94
Speech	Euclidean distance of segments from mean English and mean Spanish i-Vectors, length of turn in seconds	Linear Discriminant Analysis classifier (with SMOTE oversampling)	0.68	0.71	0.19	**0.67**
	OpenSMILE features	Linear Discriminant Analysis classifier	0.75	0.20	0.16	0.55
	Sequence of i-Vectors	LSTM[4]	0.77	0.36	0.15	0.61
Majority Baseline	N/A	N/A	0.79	0.0	0.0	0.51
Random Baseline	N/A	N/A	0.44	0.32	0.09	0.39

Table 2: Performance of Speech and Text systems (Eng:English, Spa:Spanish, CS: Code-switched). The weighted average F1 score of the best-performing text and speech systems are in bold.

	English	Codeswitched	Spanish
English	4513	8	18
Codeswitched	20	352	37
Spanish	122	7	1936

Table 3: Confusion Matrix for best-performing text system

[3] We did experiment with oversampling the code-switched class as well, but this resulted in a degradation in performance. This could probably be due to the relatively few samples in the code-switched class to begin with.

[4] We used an LSTM implementation with 200 units and a *tanh* activation function. We optimized on a categorical cross-entropy loss function using the Adam optimizer.

	English	Codeswitched	Spanish
English	3259	504	776
Codeswitched	217	97	95
Spanish	1585	1	2953

Table 4: Confusion matrix for best-performing speech system

	Segment or Turn	SMOTE?	N (train)	N (test)	English	Spanish	CS	Avg (weighted)
Monolingual	Segment	No	29,583	7,073	0.85	0.55	N/A	0.76
		Yes	39,488	10,016	0.72	0.81	N/A	0.77
	Turn	No	26,708	6,604	0.85	0.50	N/A	0.74
		Yes	36,062	9,078	0.75	0.71	N/A	0.73
3-Class	Turn	No	28,415	7,013	0.77	0.50	0.21	0.66
		Yes	37,769	9,487	0.68	0.71	0.19	0.67

Table 5: Speech system varieties

Table 2 lists the best performing text and speech systems, including the feature sets and model details. We observe that the best text system significantly outperforms the majority vote baseline by a huge margin, with an overall weighted average F1 score of 0.97 and an F1 score of 0.91 for the code-switched category. We also observe that character- and word-level features combined with a conditional random field (CRF) classifier perform slightly better than word embedding features fed into a long-short term memory neural network (LSTM).

On the other hand, LID performance dips for speech-based systems relative to their text-based counterparts for the systems we investigated, with the best performing speech system (using i-Vectors and turn length) yielding an overall weighted average F1 score of 0.67. In this case, the F1 score for the code-switched class is pretty low – 0.19. A closer examination of the confusion matrices for the best performing text and speech systems (Tables 3 and 4 respectively) provides more insight into the performance gap between the two – the text systems have a lot less confusability between classes, especially for the code-switched class. While this is not entirely surprising given that we are not directly incorporating linguistic information into our speech feature front-ends, and the large amount of far-field noise and background chatter in the speech data, this result is still in much need of improvement. Having said that, all speech systems still perform well over the majority vote baseline F1 score of 0.51.

In order to investigate to what extent the smaller sample size of the code-switched category is responsible for bringing down the LID performance of the speech-based systems, we looked at the performance of our speech systems on a 2-class (English–Spanish) classification problem at both the segment and the turn level by removing all code-switched turns from our training and test data. For the 2-class segment classification, we experimented with using the segment i-Vectors, using the Euclidean and Cosine distance between each segment and the mean vector for each class, as well as the segment-level PLDA scores for English and Spanish as features, with a the same set of learners tested on other experiments, and report the weighted average F1 for the best performing system among these. We further examined the effect of class imbalance in the Bangor corpus – recall that the corpus contains 64% English turns, 30% Spanish turns, and 6% code-switched turns – on system performance. Table 5 lists the results of these experiments. We observe that in general, the English–Spanish ”monolingual” classifiers do much better in terms of class-specific F1 scores than the ”3-class” classifiers. Using the synthetic minority oversampling technique (SMOTE) to overcome class imbalance also helped boost performance, suggesting that class imbalance and the small amounts of code-switched data contributed to a performance drop. Finally, we also hypothesize that audio quality issues in the noisy Bangor corpus speech data might have contributed to the lower numbers of the speech system. Systematically investigating this hypothesis is a subject for future research[5].

6 Discussion and Outlook

We have presented an experimental evaluation of different text and speech-based featuresets in performing language identification at the turn level in human-human dialog interactions. While the best text-based system performed excellently and at par with the state of the art in the field, the best speech-based i-Vector system did not perform as well, but still significantly outperformed the majority vote chance baseline. We observed that one of the reasons for the relatively poor performance of the speech-based system could be the relatively noisy audio that contains significant amounts of far-field and background noise. This, along with the greater percentage of English than Spanish or code-switched turns in the database, might have contributed to the performance drop. However, note that as in the case of the text-based systems, we are not directly using any linguistic or syntactic information for the speech-based systems, which undoubtedly impacts the efficacy of the latter, since such information is extremely useful for the task of language identification.

That having been said, we will explore a number of potential avenues for improving the performance of the speech-based LID systems. Chief among these, as mentioned earlier, will be to investigate techniques for noise-robustness in order to improve the LID performance of speech systems and bring them at par with their text counterparts. In addition, we would like to explore the performance of more feature–learner systems, including a more comprehensive study of deep neural network-based learners. Finally, it will be important to see how such LID systems perform on different code-switched datasets, both within and across language pairs, in order to truly test the robustness of systems across languages and dataset bias.

[5] In order to roughly test this hypothesis, we ran experiments wherein we used the relatively cleaner Fisher corpora (of both Spanish and English speech) for both training and testing. In this case, the F1 score obtained was 0.96, highlighting both the mismatch between the Fisher and Bangor corpora as well as the effect of noise in the Bangor corpus. Of course, there is the possibility that the 2-class classification of English and Spanish turns from monolingual turns in code-switched speech might pose more challenges than LID in *non*-code-switched speech. Nevertheless, while this test was not a systematic one (and hence reported only as a footnote), this clearly points toward the effect of dataset quality on system performance.

Going forward, understanding and processing code-switched speech has many implications for building code-switching dialog systems. For instance, integrating an explicit language identification step into the automatic speech recognition (ASR) module could help enhance the recognition performance. However, such solutions still require one to develop an ASR for each of the languages being analyzed. This starts becoming increasingly impractical if one wants to scale applications to multiple language pairs – an example use case is in the case of designing dialog solutions for non-native English language learners (ELLs) from multiple native language (or L1) backgrounds. In such cases, research into end-to-end spoken language understanding or SLU (where we directly go from the speech input to the SLU hypothesis) becomes very useful; and language identification would be a key component of such modules. Over and above SLU applications, such an LID module might also help inform pragmatic considerations during dialog management and the language generation module for the generation of appropriate mixed-language output. We therefore believe that in many ways this study has just scratched the surface of interesting and relevant research directions in the automated processing and modeling of code-switched dialog.

References

1. L. Milroy and P. Muysken, *One speaker, two languages: Cross-disciplinary perspectives on code-switching*. Cambridge University Press, 1995.
2. L. Wei, *The bilingualism reader*. Psychology Press, 2000.
3. J. MacSwan, "Code switching and grammatical theory," *The handbook of bilingualism*, vol. 46, p. 283, 2004.
4. C. Myers-Scotton, "Codeswitching with english: types of switching, types of communities," *World Englishes: Critical Concepts in Linguistics*, vol. 4, no. 3, p. 214, 2006.
5. B. H. OConnor and L. J. Crawford, "An art of being in between: The promise of hybrid language practices," in *Research on Preparing Inservice Teachers to Work Effectively with Emergent Bilinguals*. Emerald Group Publishing Limited, 2015, pp. 149–173.
6. R. S. Wheeler, "Code-switching," *EDUCATIONAL LEADERSHIP*, 2008.
7. Y.-L. B. Jiang, G. E. García, and A. I. Willis, "Code-mixing as a bilingual instructional strategy," *Bilingual Research Journal*, vol. 37, no. 3, pp. 311–326, 2014.
8. B. H. Ahmad and K. Jusoff, "Teachers code-switching in classroom instructions for low english proficient learners," *English Language Teaching*, vol. 2, no. 2, p. 49, 2009.
9. W. Littlewood and B. Yu, "First language and target language in the foreign language classroom," *Language Teaching*, vol. 44, no. 1, pp. 64–77, 2011.
10. T. Solorio, E. Blair, S. Maharjan, S. Bethard, M. Diab, M. Gohneim, A. Hawwari, F. Al-Ghamdi, J. Hirschberg, A. Chang *et al.*, "Overview for the first shared task on language identification in code-switched data," in *Proceedings of the First Workshop on Computational Approaches to Code Switching*. Citeseer, 2014, pp. 62–72.
11. K. Bali, Y. Vyas, J. Sharma, and M. Choudhury, "i am borrowing ya mixing? an analysis of english-hindi code mixing in facebook," *Proceedings of the First Workshop on Computational Approaches to Code Switching, EMNLP 2014*, p. 116, 2014.
12. G. Molina, N. Rey-Villamizar, T. Solorio, F. AlGhamdi, M. Ghoneim, A. Hawwari, and M. Diab, "Overview for the second shared task on language identification in code-switched data," *EMNLP 2016*, p. 40, 2016.

13. T. Solorio and Y. Liu, “Learning to predict code-switching points,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 973–981.
14. U. Barman, A. Das, J. Wagner, and J. Foster, “Code mixing: A challenge for language identification in the language of social media,” *EMNLP 2014*, vol. 13, 2014.
15. B. King and S. P. Abney, “Labeling the languages of words in mixed-language documents using weakly supervised methods.” in *HLT-NAACL*, 2013, pp. 1110–1119.
16. R. Shirvani, M. Piergallini, G. S. Gautam, and M. Chouikha, “The howard university system submission for the shared task in language identification in spanish-english codeswitching,” in *Proceedings of The Second Workshop on Computational Approaches to Code Switching*, 2016, pp. 116–120.
17. Y. Samih, S. Maharjan, M. Attia, L. Kallmeyer, and T. Solorio, “Multilingual code-switching identification via lstm recurrent neural networks,” *EMNLP 2016*, p. 50, 2016.
18. Y. Li, Y. Yu, and P. Fung, “A mandarin-english code-switching corpus.” in *LREC*, 2012, pp. 2515–2519.
19. D.-C. Lyu, T.-P. Tan, E.-S. Chng, and H. Li, “Mandarin–english code-switching speech corpus in south-east asia: Seame,” *Language Resources and Evaluation*, vol. 49, no. 3, pp. 581–600, 2015.
20. J. Y. Chan, P. Ching, and T. Lee, “Development of a cantonese-english code-mixing speech corpus.” in *INTERSPEECH*, 2005, pp. 1533–1536.
21. A. Dey and P. Fung, “A hindi-english code-switching corpus.” in *LREC*, 2014, pp. 2410–2413.
22. K. Donnelly and M. Deuchar, “The bangor autoglosser: a multilingual tagger for conversational text,” *ITA11, Wrexham, Wales*, 2011.
23. V. Ramanarayanan and D. Suendermann-Oeft, “Jee haan, I’d like both, por favor: Elicitation of a Code-Switched Corpus of Hindi–English and Spanish–English Human–Machine Dialog,” *Proc. Interspeech 2017*, pp. 47–51, 2017.
24. N. T. Vu, D.-C. Lyu, J. Weiner, D. Telaar, T. Schlippe, F. Blaicher, E.-S. Chng, T. Schultz, and H. Li, “A first speech recognition system for mandarin-english code-switch conversational speech,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4889–4892.
25. C.-F. Yeh, L.-C. Sun, C.-Y. Huang, and L.-S. Lee, “Bilingual acoustic modeling with state mapping and three-stage adaptation for transcribing unbalanced code-mixed lectures,” in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5020–5023.
26. F. Eyben, F. Weninger, F. Gross, and B. Schuller, “Recent developments in opensmile, the munich open-source multimedia feature extractor,” in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013, pp. 835–838.
27. Y. Qian, K. Evanini, X. Wang, D. Suendermann-Oeft, R. A. Pugh, P. L. Lange, H. R. Molloy, and F. K. Soong, “Improving sub-phone modeling for better native language identification with non-native english speech,” *Proc. Interspeech 2017*, pp. 2586–2590, 2017.
28. D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The kaldi speech recognition toolkit,” in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.
29. N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.
30. D. Martinez, O. Plchot, L. Burget, O. Glembek, and P. Matějka, “Language recognition in ivectors space,” in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.
31. C. Cieri, D. Miller, and K. Walker, “The fisher corpus: a resource for the next generations of speech-to-text.” in *LREC*, vol. 4, 2004, pp. 69–71.
32. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.