# Balancing the Scales: HyperSMOTE for Enhanced Hypergraph Classification

Lulwah Alkulaib* ‡, and Chang-Tien Lu*
‡ Department of Computer Science, Kuwait University, Kuwait
{lalkulaib, ctlu}@vt.edu

2023 IEEE International Conference on Big Data (BigData) | 979-8-3503-2445-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/BigData59044.2023.10386107

***Abstract*—With the proliferation of bots on social media platforms, especially X, the need for effective and efficient bot detection mechanisms has never been more paramount. However, the inherent imbalance between the number of genuine users and bots presents a significant challenge, often leading to biased classifiers. In this paper, we introduce HyperSMOTE, a novel approach for imbalanced node classification leveraging the rich structure of hypergraphs. By representing X users as nodes and their interactions as hyperedges, we construct a hypergraph that captures the intricate relationships and interactions among users. This hypergraph-based representation allows for a more nuanced understanding of user behavior and interactions, providing a robust foundation for bot detection. HyperSMOTE addresses the class imbalance by generating synthetic bot accounts in the hypergraph, ensuring a balanced training dataset while preserving the hypergraph's semantics. Our method significantly outperforms existing baselines across various metrics, demonstrating its efficacy. We further delve into the impact of different upsampling scales on classification performance, providing insights into the optimal configurations for HyperSMOTE.**

***Index Terms*—hypergraph, hypergraph learning, bot detection, class imbalance, node classification**

## I. Introduction

Online Social Networks (OSNs), particularly platforms like X (formerly known as Twitter), are pivotal in modern communication and influence public opinion [1]. However, they face challenges like misinformation and polarization, with concerns about destabilizing democratic systems [2]. A major issue is the rise of bots on X, with many acting maliciously and mimicking human behavior, complicating detection [3]. This highlights the need for advanced bot detection methods to distinguish genuine from malicious accounts.

Over the past decade, extensive research has focused on detecting X bots [4], resulting in various datasets and detection techniques. Traditional methods, based on hand-crafted features and classic machine learning [5], have faced challenges due to the evolving sophistication of X bots. Deep learning models, like recurrent neural networks and graph convolutional networks [6], have emerged to address this. However, they grapple with challenges such as: **(1) imbalanced datasets with more human than bot accounts, (2) limited labeled data in semi-supervised scenarios, and (3) previous algorithms' neglect of relationships in graph data, assuming independent samples.**

Class imbalance in node classification is a prevalent challenge in graph-based machine learning [7]. Traditional methods have tackled this issue through data-level [8], algorithm-level [9], and hybrid approaches [10]. Data-level techniques aim to balance class distribution using over-sampling or down-sampling. Algorithm-level methods introduce penalties for misclassification or set prior probabilities for classes. However, directly applying these to graphs can be suboptimal. Recently, innovative techniques have emerged that synthesize minority oversampling specifically for node classification with Graph Neural Networks (GNNs) [7], addressing the unique challenges of preserving relational information and maintaining topology structures in graphs.

The topological properties of social networks, such as X, can be structurally characterized as a hypergraph. In this representation, users serve as nodes and their relationships or interactions with one another form hyperedges. Each account on X embodies a distinct node, with its interactions, tweets, and other activities constituting the feature vector of that node. The formation of hyperedge connections between nodes is determined by the presence of interactions between accounts, such as retweets, mentions, or shared content. Given the significant disparity in the number of genuine accounts compared to bot accounts, there exists a class imbalance in the nodes of the X hypergraph. While oversampling methods offer a solution to this class imbalance problem, their direct application in a hypergraph setting presents challenges. One of the primary difficulties is establishing the connections between synthetic samples and the original hypergraph while ensuring the preservation of the hypergraph's integrity and semantics.

The need to solve the challenges mentioned above and the wide range of applications with imbalanced data classes motivated us to propose a hypergraph neural network for imbalanced node classification. The objective of this work is to create a robust system that evaluates X accounts while addressing challenges like dataset imbalance by synthesizing minority oversampling for node classification with hypergraphs, ensuring the preservation of relational information, capturing high-order relations, and maintaining the integrity of topology structures.

In summary, our contributions are as follows:

1) **Hypergraph Representation of X Ecosystem:** We introduce a novel representation of the X platform as a hypergraph, where each user account is depicted as a node and their interactions (tweets, retweets, mentions, etc.) form hyperedges. This representation captures the intricate relationships and interactions between X ac-

counts, providing a rich structure for subsequent analysis and bot detection.

2) **Addressing Class Imbalance with HyperSMOTE:** Recognizing the significant class imbalance in the hypergraph, where genuine user accounts outnumber bot accounts, we propose the HyperSMOTE technique. This method generates synthetic bot accounts in the hypergraph, ensuring a balanced training dataset while preserving the hypergraph's structure and semantics.
3) **Hypergraph Neural Network for Bot Detection:** We design a hypergraph neural network that aggregates information from neighboring nodes and hyperedges to generate a robust representation for each node. The network is trained to classify nodes (X accounts) as either bots or genuine users, with the aim of achieving high accuracy in bot detection.
4) **Empirical Validation:** We perform comprehensive experiments centered on bot detection in the X hypergraph. The empirical results highlight the effectiveness of our proposed methods, demonstrating their superiority in detecting bots, especially in scenarios with pronounced class imbalances.

## II. Related Work

### A. Bot Detection

The rapid proliferation of bots on X has necessitated advanced detection techniques. Traditional supervised methods relied on annotated datasets, but their static nature made them susceptible to evolving bot behaviors [5], [11], [12]. Recent approaches have shifted towards graph techniques, with methods like SATAR [13] and GCN-based approaches [6] leveraging network structures. However, their efficacy is closely tied to specific datasets and bot types. Deep learning, employing architectures like GANs [14] and RNNs [15], has shown promise in bot detection, albeit with the challenge of being "black-box" models. The dynamic landscape of bot detection indicates the need for continuous research and adaptation.

### B. Class Imbalance in Graph-based Methods

In graph-based machine learning, the class imbalance problem is a prevalent challenge. Generally, there are two primary techniques to address this issue: algorithm-level and data-level approaches. Algorithm-level methods modify classifiers by adjusting specific mechanisms that might falter in imbalanced settings. However, the efficacy of these modified classifiers is often contingent upon the performance of the chosen classifier, which can be limiting in certain scenarios [16]–[18]. On the other hand, data-level techniques aim to balance class distribution using over-sampling or down-sampling. Directly applying these techniques to graphs can be suboptimal due to the unique challenges of preserving relational information and maintaining topology structures. Graph Neural Networks (GNNs) have emerged as a promising solution to address these challenges [19]. For instance, graphSMOTE [20] synthesizes minority samples in graph-structured data, while GATSMOTE [21] extends the GAT architecture to tackle imbalanced node classification. By leveraging the inherent structure of graph data, GNNs can capture complex relationships between nodes, making them particularly suited for tasks like imbalanced node classification. However, the application of GNNs in this domain requires innovative techniques that can synthesize minority oversampling while ensuring the preservation of graph topology and relational information. Despite the advancements, there's a growing realization that hypergraphs, with their ability to capture higher-order relationships, could offer a more comprehensive representation. Incorporating hypergraphs into imbalance correction techniques might address the nuances that traditional graph-based methods might overlook, paving the way for more robust solutions.

### C. Hypergraph Learning

The realm of hypergraph learning has recently seen a surge in interest, especially in areas such as classification [22], link prediction [23], and community detection [24]. While traditional graphs offer a foundational understanding, hypergraphs provide a richer representation by capturing intricate relationships and high-order interactions between data points. This advanced modeling allows hypergraphs to delve into a more expansive nonlinear space, enhancing their ability to represent complex correlations and subsequently boosting their performance in various tasks [25]. In the context of our study, we harness the power of hypergraphs to intricately map the dynamic ecosystem of X, which includes users, their tweets, and the complex web of interactions between them.

## III. Proposed Method

In this section, we formally define the problem statement and then introduce our proposed approach, HyperSMOTE, a hypergraph-based learning method to detect bot accounts in an X's sub-graph using oversampling. Figure 1 illustrates the difference between oversampling methods.

### A. Problem Definition

Given the X ecosystem, we represent each user account as a node and their interactions (tweets, retweets, mentions, etc.) as hyperedges, forming a hypergraph. Let's denote this hypergraph as $\mathcal{H} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes (X accounts) and $\mathcal{E}$ is the set of hyperedges (interactions between accounts). Each node $n \in \mathcal{N}$ has an associated label $y_n$, where $y_n = 1$ indicates a bot account and $y_n = 0$ indicates a genuine user account. Our objective is to design a classifier $f : \mathcal{N} \rightarrow \{0, 1\}$ that predicts the label of each node based on its features and its position within the hypergraph. The challenge lies in the imbalanced nature of the dataset, where genuine user accounts significantly outnumber bot accounts. This imbalance can lead to biased classifiers that perform poorly on the minority class (bots). To address this, we propose a hypergraph-based SMOTE approach to generate synthetic bot accounts, ensuring a balanced training dataset while preserving the intricate structure and semantics of the hypergraph.

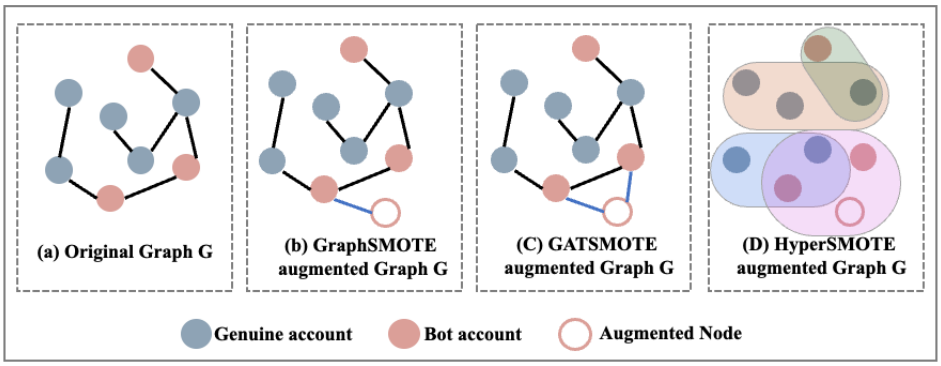



Fig. 1. Illustrating the difference between oversampling methods

### B. Hypergraph Construction

A hypergraph is a generalization of a graph where an edge, termed a hyperedge, can connect any number of nodes. In the context of X, this allows us to capture more complex relationships and interactions between user accounts.

*1) Node Representation:* Each user account on X is represented as a node $n$ in the hypergraph. Let $\mathcal{N}$ be the set of all nodes, where each node $n_i$ has an associated feature vector $\mathbf{x}_i$ that encapsulates information about the user's profile, tweeting behavior, and other relevant attributes.

*2) Hyperedge Formation:* Hyperedges in our hypergraph are formed based on interactions between user accounts. We define three primary types of hyperedges:

1. **Tweet Interaction Hyperedges:** For every tweet, the author of the tweet and all users who interact with it (e.g., retweets, replies, mentions) form a hyperedge. Mathematically, for a tweet $t_j$, the hyperedge $e_{t_j}$ is defined as:

$$e_{t_j} = \{n | n \text{ interacts with } t_j\} \tag{1}$$

2. **Content Sharing Hyperedges:** Users who share similar content or hashtags can be grouped under a hyperedge. For a specific content or hashtag $h_k$, the hyperedge $e_{h_k}$ is:

$$e_{h_k} = \{n | n \text{ shares content or hashtag } h_k\} \tag{2}$$

3. **Temporal Interaction Hyperedges:** Users who interact within a specific time window can be connected by a hyperedge, capturing temporal patterns of interaction. For a time window $\tau_l$, the hyperedge $e_{\tau_l}$ is:

$$e_{\tau_l} = \{n | n \text{ interacts within time window } \tau_l\} \tag{3}$$

The set of all hyperedges is denoted as $\mathcal{E}$. Each hyperedge $e \in \mathcal{E}$ is associated with a weight $w_e$ that signifies the strength or importance of that hyperedge in the hypergraph.

*3) Hypergraph Weighting:* The weights of the hyperedges can be determined based on the frequency of interactions, the significance of shared content, or other relevant metrics. Formally, the weight $w_e$ of a hyperedge $e$ can be computed as:

$$w_e = \frac{\sum_{n \in e} \text{interaction frequency of } n}{|e|} \tag{4}$$

where $|e|$ denotes the number of nodes in hyperedge $e$.

This construction ensures that the hypergraph captures the intricate relationships and interactions between X accounts, providing a rich structure for subsequent analysis and bot detection.

*4) Addressing Class Imbalance with HyperSMOTE:* Given the imbalanced nature of the dataset, where bot accounts are outnumbered by genuine accounts, directly training the hypergraph neural network can lead to a biased model that predominantly classifies accounts as genuine. To address this, we employ the HyperSMOTE technique according to Algorithm 1 to generate synthetic nodes (accounts) in the hypergraph.

**Algorithm 1** HyperSMOTE for Hypergraph Node Augmentation

**Require:** Hypergraph $\mathcal{H} = (\mathcal{N}, \mathcal{E})$, Minority node set $\mathcal{N}_{min}$, Oversampling rate $r$
**Ensure:** Augmented hypergraph $\mathcal{H}'$
**for** each bot node $n_i$ in $\mathcal{N}_{min}$ **do**
  **for** $j = 1$ to $r$ **do**
    Identify neighboring nodes of $n_i$ and associated hyperedges
    Randomly select a neighboring bot node $n_j$
    Compute the difference in feature vectors: $\Delta\mathbf{x} = \mathbf{x}_{n_j} - \mathbf{x}_{n_i}$
    Generate a synthetic feature vector: $\mathbf{x}_{syn} = \mathbf{x}_{n_i} + \lambda\Delta\mathbf{x}$ where $\lambda$ is a random number between 0 and 1
    Add the synthetic node with feature vector $\mathbf{x}_{syn}$ to $\mathcal{N}$
    Connect the synthetic node with relevant hyperedges based on similarity criteria
  **end for**
**end for**
  **return** Augmented hypergraph $\mathcal{H}' = (\mathcal{N}, \mathcal{E})$

This process augments the dataset with synthetic bot accounts, ensuring a more balanced distribution between bot and genuine accounts. The synthetic nodes are generated in a manner that respects the hypergraph structure, ensuring that the topological properties and relationships are preserved.

### C. Hypergraph Neural Network for Bot Detection

Given the constructed hypergraph, our goal is to learn a function that can classify nodes (X accounts) as either bots or genuine users. To achieve this, we propose a hypergraph neural network that aggregates information from neighboring nodes and hyperedges to generate a robust representation for each node.

*1) Hypergraph Convolution:* The convolution operation on a hypergraph is an extension of graph convolution. For each node $n_i$, we aggregate information from its neighboring nodes and the hyperedges it is associated with. The aggregated feature for node $n_i$ is given by:

$$\mathbf{h}_{n_i} = \sigma\left(\sum_{e\in\mathcal{E}} w_e \sum_{n_j\in e} \mathbf{A}_{ij}\mathbf{W}\mathbf{x}_{n_j}\right) \tag{5}$$

where $\mathbf{A}$ is the adjacency tensor of the hypergraph, $\mathbf{W}$ is a learnable weight matrix, and $\sigma$ is a non-linear activation function.

*2) Pooling and Classification:* After convolution, we apply a pooling layer to down-sample the node features and capture the most important information. The pooled feature matrix $\mathbf{P}$ can be obtained using various pooling strategies such as mean pooling or max pooling.

Finally, the pooled features are passed through a fully connected layer followed by a softmax activation to obtain the classification probabilities for each node:

$$\mathbf{y}_{n_i} = \text{softmax}(\mathbf{P}\mathbf{W}_c + \mathbf{b}_c) \tag{6}$$

where $\mathbf{W}_c$ and $\mathbf{b}_c$ are the weight matrix and bias vector for the classification layer, respectively.

*3) Loss Function and Training:* We employ the cross-entropy loss for training our model. Given the true labels $\mathbf{Y}$ and the predicted probabilities $\mathbf{Y}_{pred}$, the loss $\mathcal{L}$ is:

$$\mathcal{L} = -\sum_{i=1}^{N} \mathbf{Y}_i \log(\mathbf{Y}_{pred,i}) \tag{7}$$

where $N$ is the number of nodes.

The model is trained using gradient descent optimization algorithms, adjusting the weights to minimize the loss and improve the classification accuracy.

*4) Bot Detection:* After training, for a given node $n_i$, if $\mathbf{y}_{n_i}[1] > \theta$, where $\theta$ is a threshold, the node is classified as a bot. Otherwise, it is classified as a genuine user. The threshold $\theta$ can be set based on the desired trade-off between precision and recall.

## IV. Experiments

### A. Dataset

We evaluate our proposed method on three real-world X bot dataset benchmarks. The dataset description can be found in Table I, where each dataset is described as follows:

- **verified-2019 [26]**: A collection of authenticated human X accounts, showcasing genuine user behaviors and interactions.

TABLE I
DATASETS DETAILS

| | verified-2019 & botwiki-2019 | cresci-rtbust-2019 |
|---|---|---|
| # Nodes | 53,321 | 824,902 |
| # Bots | 15,996 | 148,482 |
| # Humans | 37,325 | 676,420 |
| # Attributes | 17,509 | 42,051 |

- **botwiki-2019 [26]**: A set of self-declared bot accounts, offering insights into bot strategies. For our study, it's merged with *verified-2019* [26] to create an imbalanced human vs. bot dataset.
- **cresci-rtbust-2019 [27]**: A dataset capturing Italian user activities over two weeks in 2018, highlighting retweet patterns and bot interactions.

### B. Baselines

We compare our proposed method, HyperSMOTE, against four existing bot user detection baselines and four class-imbalance baselines:

- **Botometer** [28]: A widely-used tool that evaluates the likelihood of a Twitter account being a bot. It leverages a combination of machine learning techniques and analyzes various account features, including tweet metadata, sentiment analysis, and network patterns.
- **SATAR** [13]: A method that combines user information with Twitter network graph structure features to identify bots. It captures both content and connection patterns to differentiate between genuine users and automated accounts.
- **BotRGCN** [29]: Utilizes relational graph convolutional networks to represent the Twitter network and user features. By capturing the relational dependencies between users, it offers a robust framework for bot detection.
- **ADNET** [30]: An anomaly detection network designed to identify unusual patterns in user behavior. By treating bot detection as an anomaly detection task, ADNET can identify bots that exhibit behaviors deviating from typical user patterns.
- **SMOTE** [7]: Synthetic Minority Over-sampling Technique is a popular method to address the class imbalance by generating synthetic samples in the feature space to balance out the minority class.
- **Embed-SMOTE** [31]: An extension of SMOTE that leverages embeddings to generate synthetic samples. It captures the semantic relationships in the data, ensuring that the synthetic samples are contextually relevant.
- **GraphSMOTE** [20]: Adapts the SMOTE technique for graph data. It synthesizes nodes in a way that respects the graph structure, ensuring that the topological properties of the graph are preserved.
- **GATSMOTE** [21]: Combines the Graph Attention Networks (GAT) with SMOTE to address the class imbalance in graph data. It leverages attention mechanisms to weigh

the importance of neighboring nodes when generating synthetic samples.

### C. Evaluation Metrics

We utilize several metrics to assess bot detection performance. Accuracy measures the overall correctness, while precision and recall evaluate the detection of actual bots. The $F_1$ score offers a balance between precision and recall. Additionally, the AUC-ROC assesses the model's distinguishing ability between bot and genuine accounts.

## V. Results

Fig. 2. Accuracy for different Imbalance Ratios.

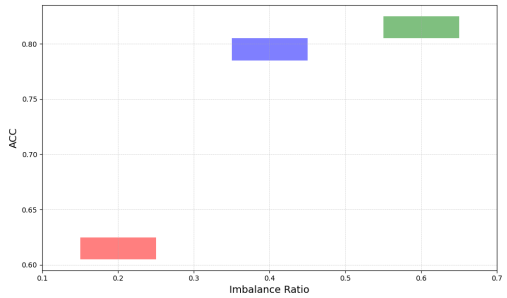
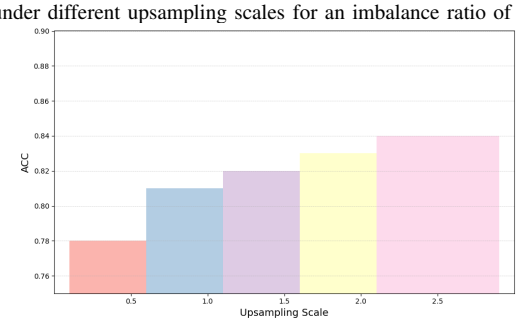



Fig. 3. Accuracies of proposed HyperSMOTE for imbalanced node classification under different upsampling scales for an imbalance ratio of 0.5.




In our approach, the hypergraph structure was leveraged to represent intricate relationships and interactions between X accounts, capturing both the content and connection patterns. By introducing HyperSMOTE, we aimed to address the class imbalance inherent in bot detection tasks, generating synthetic bot accounts that respect the hypergraph's structure and semantics. Our proposed HyperSMOTE method consistently surpassed existing baselines across both datasets, as shown in Table II, underscoring the efficacy of our model. We delve deeper into the performance and nuances of each component of our approach in the following:

1) **Superiority of HyperSMOTE**: The most striking observation is the unparalleled performance of HyperSMOTE. On the "verified-2019 & botwiki-2019" dataset, HyperSMOTE achieved an accuracy (ACC) of 0.815, an F1 score of 0.809, and an AUC-ROC of 0.957. Similarly, for the "cresci-rtbust-2019" dataset, HyperSMOTE's performance metrics were 0.844, 0.845, and 0.970, respectively. These figures are notably higher than any other method, underscoring the effectiveness of HyperSMOTE in addressing class imbalances in node classification tasks.
2) **Comparison with GATSMOTE**: GATSMOTE, another prominent method, was the second-best performing model, particularly with an accuracy of 0.727 and an AUC-ROC of 0.895 on the first dataset. However, even with its robust performance, it was still outperformed by HyperSMOTE.
3) **Performance of Traditional Methods**: Traditional methods like Botometer and SATAR delivered consistent results across both datasets. For instance, Botometer achieved accuracy values of 0.637 and 0.606 for the two datasets, respectively. However, these figures were considerably lower than the more advanced techniques like HyperSMOTE and GATSMOTE.
4) **Emergence of SMOTE Variants**: It's evident that various adaptations of the SMOTE technique, such as Embed-SMOTE and GraphSMOTE, have improved the performance metrics compared to the original SMOTE. Especially, GraphSMOTE's accuracy of 0.725 on the "verified-2019 & botwiki-2019" dataset indicates the potential of adapting traditional techniques to more complex structures like hypergraphs.
5) **Imbalance Ratio Performance**: As shown in Figure 2The HyperSMOTE method demonstrates increasing accuracy as the imbalance ratio rises, highlighting its effectiveness and adaptability in handling varying degrees of dataset imbalance.
6) **Performance Enhancement with Upsampling**: From Figure 3, we notice HyperSMOTE's accuracy consistently improves as the upsampling scale increases, indicating the method's capability to benefit from additional synthetic data.
7) **Saturation Beyond a Point**: Additionally, Figure 3 shows that while there is a noticeable increase in accuracy from an upsampling scale of 0.5 to 1.5, the gains start to diminish slightly beyond this point, suggesting a potential saturation or optimal point for upsampling in this context.

In summary, while several methods showcased commendable results in imbalanced node classification, HyperSMOTE's performance stood out, emphasizing its potential as a leading technique in the class imbalance domain.

## VI. Conclusion

In this study, we presented HyperSMOTE, a novel approach for imbalanced node classification within hypergraphs tailored for the X ecosystem. By harnessing the intricate structure of hypergraphs, HyperSMOTE consistently surpassed existing baselines across various datasets and metrics. Our findings highlight the significance of selecting the appropriate upsampling scale and the adaptability of HyperSMOTE to diverse imbalance ratios. Ultimately, HyperSMOTE emerges as a promising tool for bot detection in social media, emphasizing the potential of hypergraph-based techniques in addressing classification challenges in imbalanced scenarios.

TABLE II
IMBALANCED NODE CLASSIFICATION PERFORMANCE

| Dataset | verified-2019 & botwiki-2019 | | | cresci-rtbust-2019 | | |
|---|---|---|---|---|---|---|
| **Metric** | **ACC** | **F1** | **AUC-ROC** | **ACC** | **F1** | **AUC-ROC** |
| Botometer | 0.637 | 0.640 | 0.856 | 0.606 | 0.599 | 0.813 |
| SATAR | 0.648 | 0.623 | 0.859 | 0.616 | 0.583 | 0.817 |
| BotRGCN | 0.652 | 0.640 | 0.869 | 0.620 | 0.599 | 0.826 |
| ADNET | 0.651 | 0.630 | 0.861 | 0.619 | 0.590 | 0.819 |
| SMOTE | 0.639 | 0.630 | 0.855 | 0.608 | 0.590 | 0.813 |
| Embed-SMOTE | 0.689 | 0.680 | 0.874 | 0.655 | 0.637 | 0.831 |
| GraphSMOTE | 0.725 | 0.724 | 0.882 | 0.690 | 0.677 | 0.838 |
| GATSMOTE | 0.727 | 0.723 | 0.895 | 0.692 | 0.676 | 0.851 |
| **HyperSMOTE** | **0.815** | **0.809** | **0.957** | **0.844** | **0.845** | **0.970** |

## REFERENCES


[1] B. E. Weeks, A. Ardèvol-Abreu, and H. Gil de Zúñiga, "Online influence? social media use, opinion leadership, and political persuasion," *International journal of public opinion research*, vol. 29, no. 2, pp. 214–239, 2017.

[2] D. A. Broniatowski, A. M. Jamison, S. Qi, L. AlKulaib, T. Chen, A. Benton, S. C. Quinn, and M. Dredze, "Weaponized health communication: Twitter bots and russian trolls amplify the vaccine debate," *American journal of public health*, vol. 108, no. 10, pp. 1378–1384, 2018.

[3] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Who is tweeting on twitter: human, bot, or cyborg?," in *Proceedings of the 26th annual computer security applications conference*, pp. 21–30, 2010.

[4] T. Khaund, B. Kirdemir, N. Agarwal, H. Liu, and F. Morstatter, "Social bots and their coordination during online campaigns: A survey," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 2, pp. 530–545, 2022.

[5] O. Varol, C. A. Davis, F. Menczer, and A. Flammini, "Feature engineering for social bot detection," in *Feature engineering for machine learning and data analytics*, pp. 311–334, CRC Press, 2018.

[6] S. Feng, H. Wan, N. Wang, and M. Luo, "Botrgcn: Twitter bot detection with relational graph convolutional networks," in *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 236–239, 2021.

[7] 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.

[8] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *International journal of pattern recognition and artificial intelligence*, vol. 23, no. 04, pp. 687–719, 2009.

[9] V. S. Spelmen and R. Porkodi, "A review on handling imbalanced data," in *2018 international conference on current trends towards converging technologies (ICCTCT)*, pp. 1–11, IEEE, 2018.

[10] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases: PKDD 2003: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003. Proceedings 7*, pp. 107–119, Springer, 2003.

[11] O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini, "Online human-bot interactions: Detection, estimation, and characterization," in *Eleventh international AAAI conference on web and social media*, 2017.

[12] K.-C. Yang, O. Varol, C. A. Davis, E. Ferrara, A. Flammini, and F. Menczer, "Arming the public with artificial intelligence to counter social bots," *Human Behavior and Emerging Technologies*, vol. 1, no. 1, pp. 48–61, 2019.

[13] S. Feng, H. Wan, N. Wang, J. Li, and M. Luo, "Satar: A self-supervised approach to twitter account representation learning and its application in bot detection," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 3808–3817, 2021.

[14] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Communications of the ACM*, vol. 59, no. 7, pp. 96–104, 2016.

[15] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[16] P. Cao, D. Zhao, and O. Zaiane, "An optimized cost-sensitive svm for imbalanced data learning," in *Pacific-Asia conference on knowledge discovery and data mining*, pp. 280–292, Springer, 2013.

[17] G. Wu and E. Y. Chang, "Class-boundary alignment for imbalanced dataset learning," in *ICML 2003 workshop on learning from imbalanced data sets II, Washington, DC*, pp. 49–56, 2003.

[18] M. Sahare and H. Gupta, "A review of multi-class classification for imbalanced data," *International Journal of Advanced Computer Research*, vol. 2, no. 3, p. 160, 2012.

[19] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[20] T. Zhao, X. Zhang, and S. Wang, "Graphsmote: Imbalanced node classification on graphs with graph neural networks," in *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 833–841, 2021.

[21] Y. Liu, Z. Zhang, Y. Liu, and Y. Zhu, "Gatsmote: Improving imbalanced node classification on graphs via attention and homophily," *Mathematics*, vol. 10, no. 11, p. 1799, 2022.

[22] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," *Advances in neural information processing systems*, vol. 19, 2006.

[23] L. Xia, C. Huang, Y. Xu, P. Dai, L. Bo, X. Zhang, and T. Chen, "Link prediction in social networks based on hypergraph," *WWW 2013 Companion - Proceedings of the 22nd International Conference on World Wide Web*, pp. 41–42, 2013.

[24] Z. T. Ke, F. Shi, and D. Xia, "Community detection for hypergraph networks via regularized tensor power iteration," 9 2019.

[25] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, and C. Zou, "Hypergraph learning: Methods and practices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[26] K.-C. Yang, O. Varol, P.-M. Hui, and F. Menczer, "Scalable and generalizable social bot detection through data selection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 1096–1103, 2020.

[27] M. Mazza, S. Cresci, M. Avvenuti, W. Quattrociocchi, and M. Tesconi, "Rtbust: Exploiting temporal patterns for botnet detection on twitter," in *Proceedings of the 10th ACM conference on web science*, pp. 183–192, 2019.

[28] C. A. Davis, O. Varol, E. Ferrara, A. Flammini, and F. Menczer, "Botornot: A system to evaluate social bots," in *Proceedings of the 25th international conference companion on world wide web*, pp. 273–274, 2016.

[29] S. Feng, H. Wan, N. Wang, and M. Luo, "Botrgcn: Twitter bot detection with relational graph convolutional networks," in *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 236–239, 2021.

[30] L. Alkulaib, L. Zhang, Y. Sun, and C.-T. Lu, "Twitter bot identification: An anomaly detection approach," in *2022 IEEE International Conference on Big Data (Big Data)*, pp. 3577–3585, IEEE, 2022.

[31] S. Ando and C. Y. Huang, "Deep over-sampling framework for classifying imbalanced data," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 10*, pp. 770–785, Springer, 2017.