USING MACHINE LEARNING ALGORITHMS TO DETECT BOTS ON E-COMERCE WEBSITES

Yohannes Geleta, Jingnan Xie

Department of Computer Science, Millersville University

{ysgeleta,jingnam.xie}@millersvile.edu

ABSTRACT

In the constantly changing landscape of online shopping, the rise of automated bots poses a substantial challenge to the reliability of e-commerce platforms. This research aspires to tackle this issue by developing a robust machine learning algorithm, utilizing decision trees, random forests, and k-Nearest Neighbors for the detection of bots in web server logs. Additionally, it also compares the difference in success of these algorithms, as well as explains why. The motivation stems from the disruptive impact of bots on user experiences and the potential threats they pose to server traffic.

Our research begins with a comprehensive dataset from the Iranian e-commerce hub zanbil.ir, encompassing extensive web server logs. Initially, the application of decision trees to a smaller dataset yields promising results, achieving 100% accuracy. However, as we scale our analysis to a more comprehensive dataset, the limitations of a standalone decision tree become apparent, leading us to investigate alternative methods. The research then delves into the intricacies of random forests, demonstrating their effectiveness in mitigating overfitting and improving accuracy, ultimately achieving a commendable 63%. Subsequently, KNN is explored for its straightforward approach, achieving an observable accuracy rate of 46%.

KEYWORDS
Decision Trees, Bots, User Agents, Random Forest Trees, KNN

1. Introduction

In the constantly changing world of online shopping, technology has transformed how we engage in commerce. However, amidst the convenience, a significant challenge has emerged—the infiltration of automated bots. These elusive digital adversaries not only disrupt user experiences but also pose a threat to the reliability of server traffic, jeopardizing the core of e-commerce platforms. So, research on bot detection becomes important and popular (for example, see [1-4]).

This paper embarks on the development of a robust machine learning algorithm, employing the power of decision trees, random forests, and k-Nearest Neighbors as other research has shown these algorithms to be effective [5]. Our aim is not just to uncover these stealthy bots but also to explore why different methods of detection work better than others.

The impact of bots in e-commerce goes beyond mere inconvenience, intertwining with the complexity of managing server traffic [9]. Real-life examples, like the race for tickets to a Taylor Swift concert on Ticketmaster, show the urgency of addressing this challenging issue. Beyond compromising the fairness of the ticketing system, the automated scalping executed by bots adds stress to server resources, impacting the overall performance and responsiveness of the entire platform. These disruptions not only hinder the smooth operation of e-commerce websites but also contribute to tangible economic losses, eroding the trust and satisfaction of genuine users.

In this research, our focus extends beyond merely identifying bot activities. We delve into applying decision trees, random forests, and k-Nearest Neighbors. Decision trees provide a clear framework for understanding decision-making, random forests offer strength through ensemble learning, and k-Nearest Neighbors bring a proximity-based approach to understanding patterns in user interactions. The integration of these methodologies seeks to create a human-tailored system that not only identifies and flags potential bot activities but also acts as a shield, strengthening the intricate web of server traffic against the disruptive impacts of digital adversaries.

Our goal is to provide e-commerce platforms with a solution that goes beyond conventional bot detection. Through the strategic use of decision trees, random forests, and k-Nearest Neighbors, we aim to enhance the resilience and reliability of e-commerce infrastructures. This effort aims to create an environment where fair competition thrives, and users can navigate the digital marketplace with

confidence and ease. This exploration underscores the potential of combining human insight with machine learning techniques to tackle the intricate challenges posed by bots in the ever-evolving landscape of e-commerce.

2. The Dataset

Our research on identifying bots begins with a robust dataset of web server logs sourced from the Iranian e-commerce hub, zanbil.ir. This dataset at a substantial 3.3 gigabytes, is a collection of logs stemming from the variety of events unfolding on the web server. These logs paint a vivid picture, capturing many different visitor behaviors, the interactivity of web crawlers, and the flow of activities on the website. Beyond just data, these logs offer invaluable business insights, user behavior nuances, and potential security red flags. In its unfiltered log file format, the dataset authentically mirrors the sheer magnitude and complexity of the logs that define the online landscape.

The "Online Shopping Store - Web Server Logs" dataset is crucial for our efforts to create an efficient pipeline for handling, parsing, compressing, and deciphering web server log files. Given that these logs are essentially the backbone of the online experience, particularly in the bustling world of e-commerce, our research deals with the challenges and opportunities inherent in managing such extensive log files. Our goal is not only to be able to detect bots with algorithms but to find out why some algorithms succeed and others don't.

In our research paper focused on deploying Decision Trees, Random Forests, and k-Nearest Neighbors for bot detection, the journey began with meticulous data preprocessing to extract insights from a vast web server log dataset. Sourced from the Iranian e-commerce website zanbil.ir and provided by Farzin Zaker, through the Harvard Dataverse, the dataset initially presented a raw and unstructured landscape that required careful curation. Initially distributed in log format, the dataset underwent a systematic transformation into a more manageable and structured CSV format. This transformation included parsing log entries, extracting relevant information, and consolidating them into a tabular structure. This critical step laid the foundation for subsequent analysis, ensuring a more streamlined and accessible dataset for our machine-learning models.

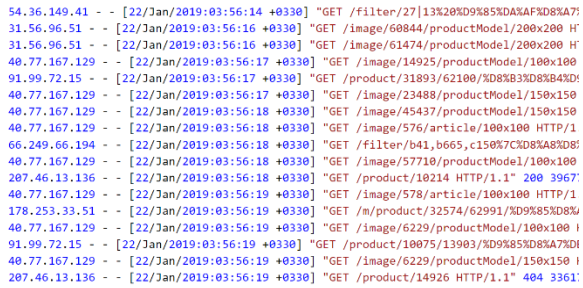

Figure 1. Log file of the original dataset

Another pivotal facet of our data preparation process involved the creation of a target variable essential for bot detection. Leveraging information encoded within the 'user_agent', 'client', and 'referer' columns, we devised an approach to categorize user agents as either human or robot. By scrutinizing the distinctive patterns in user-agent strings associated with various browsers and automated agents, we crafted a binary target variable classifying instances as either human or bot. This target creation process aimed to imbue our models with the capability to discern between human-driven interactions and those originating from automated bots.

client	userid	datetime	method	request	status	size	referer	user_agen	Bot
54.36.149	-	22/Jan/20	GET	/filter/27|	200	30577	-	Mozilla/5.(	TRUE
31.56.96.5	-	22/Jan/20	GET	/image/60	200	5667	https://ww	Mozilla/5.(	FALSE
31.56.96.5	-	22/Jan/20	GET	/image/61	200	5379	https://ww	Mozilla/5.(	FALSE
40.77.167	-	22/Jan/20	GET	/image/14	200	1696	-	Mozilla/5.(	TRUE
91.99.72.1	-	22/Jan/20	GET	/product/3	200	41483	-	Mozilla/5.(	FALSE
40.77.167	-	22/Jan/20	GET	/image/23	200	2654	-	Mozilla/5.(	TRUE
40.77.167	-	22/Jan/20	GET	/image/45	200	3688	-	Mozilla/5.(	TRUE
40.77.167	-	22/Jan/20	GET	/image/57	200	14776	-	Mozilla/5.(	TRUE
66.249.66	-	22/Jan/20	GET	/filter/b41,	200	34277	-	Mozilla/5.(	TRUE
40.77.167	-	22/Jan/20	GET	/image/57	200	1695	-	Mozilla/5.(	TRUE
207.46.13	-	22/Jan/20	GET	/product/1	200	39677	-	Mozilla/5.(	TRUE
40.77.167	-	22/Jan/20	GET	/image/57	200	9831	-	Mozilla/5.(	TRUE
178.253.3	-	22/Jan/20	GET	/m/produc	200	20406	https://ww	Mozilla/5.(	FALSE
40.77.167	-	22/Jan/20	GET	/image/62	200	1796	-	Mozilla/5.(	TRUE
91.99.72.1	-	22/Jan/20	GET	/product/1	200	41725	-	Mozilla/5.(	FALSE

Figure 2. CSV file with target 'Bot' attribute

The transformation of raw logs into a structured CSV format and the strategic creation of a target variable not only facilitated more effective feature engineering but also paved the way for a robust evaluation of the performance of Decision Trees, Random Forests, and k-Nearest Neighbors in the domain of bot detection.

3. Analysis and Results

In the dynamic field of machine-learning, the choice of algorithms plays a pivotal role in determining the success of a classification task. Our research delves into the comparative analysis of three distinct algorithms Decision Trees, Random Forests, and k-Nearest Neighbors to discern their respective strengths and weaknesses within varying contexts. As we formulate our hypotheses, we aim to discover insights into the performance of each algorithm,

considering their inherent characteristics and adaptability to diverse datasets.

Firstly, we suggest that the Decision Tree algorithm, known for its capacity to unravel non-linear relationships and offer interpretability, may shine brightest when confronted with datasets characterized by well-defined and straightforward decision boundaries. The transparent structure of Decision Trees empowers users with an intuitive understanding of the decision-making process, making it particularly advantageous in scenarios where model interpretability is vital. However, we cautiously anticipate that the interpretative ability of Decision Trees may dwindle in more intricate datasets or those where the relationships between features are complex.

Conversely, our hypothesis regarding the KNN algorithm is tinged with a sense of caution. While KNN excels in simplicity and ease of implementation, its reliance on distance metrics makes it susceptible to the curse of dimensionality. In high-dimensional feature spaces, the efficacy of KNN might diminish due to the increased influence of irrelevant or noisy features. Additionally, KNN's computational cost grows exponentially with larger datasets, potentially impeding its scalability and efficiency. Our conjecture, therefore, is that KNN may face challenges in datasets where a myriad of features and extensive data volumes are prevalent. However research has shown that KNN has been very effective at detecting bots with a 98% accuracy[8].

In contrast, our expectations for the Random Forests algorithm are optimistic. This ensemble method, which amalgamates predictions from multiple Decision Trees, holds promise in overcoming the limitations of its singular counterpart. We hypothesize that Random Forests will emerge as the frontrunner among the trio, especially in datasets characterized by intricate decision boundaries and a multitude of features. By aggregating diverse perspectives from numerous trees, Random Forests can mitigate overfitting and enhance predictive accuracy. Their ensemble nature allows them to capture a broader spectrum of patterns and dependencies within the data. Furthermore, the parallelization capabilities of Random Forests contribute to their scalability, enabling them to efficiently handle larger datasets, a potential limitation of the KNN algorithm.

To substantiate these hypotheses, our research will employ rigorous empirical evaluation of diverse datasets. By exploring how each algorithm navigates through the complexities of real-world datasets, we aim to draw practical insights for practitioners seeking optimal solutions in their specific domains. Through this comprehensive analysis, our research endeavors to illuminate the distinct merits and limitations of each algorithm, fostering a more informed and nuanced understanding within the machine learning community.

3.1 Decision Tree

In our research journey exploring machine learning algorithms for detecting bots in web server logs, we initially turned to the Decision Tree classifier. Decision trees have been proven to be very effective in detecting bots [7]. Our dataset, sourced from the web server logs of an Iranian e-commerce site (zanbil.ir), featured key elements like 'client,' 'datetime,' 'user_agent,' and a 'Bot' label indicating whether a visitor was a bot. The Decision Tree, resembling a hierarchical decision-making tool, processed the dataset by recursively splitting it based on selected features, assigning a 'Bot' or 'non-Bot' label to each leaf node. In our early work with a small dataset slice (around 0.1% of the total), the Decision Tree showcased its prowess with an impressive 100% accuracy, effectively capturing discernible patterns within this limited scope.

However, as we extended our dataset to encompass a more comprehensive view of the web server logs, the Decision Tree grappled with maintaining its earlier accuracy levels. The complexities introduced by a larger dataset, encompassing a diverse range of user agents and client details, presented challenges for the Decision Tree, and thus decreased our accuracy to 40%. This decline in accuracy highlighted the necessity for a more sophisticated approach capable of extracting nuanced patterns within the expanded dataset.

To enhance the model's adaptability to the entire dataset, we implemented one-hot encoding for categorical features such as 'client' and 'user_agent.' While this encoding effectively transformed categorical variables into a machine-learning-friendly format, the Decision Tree struggled to derive meaningful insights from the extensive logs. These challenges emphasized the limitations of relying solely on a single Decision Tree to handle the intricacies and scale of the dataset, prompting us to explore alternatives like Random Forest and k-Nearest Neighbors for more robust bot detection in diverse and extensive web server logs.

3.2. Random Forest

In our pursuit to refine the art of bot detection within the complexity of web server logs, we opted for a strategic shift, turning our attention towards the Random Forest algorithm. This ensemble learning technique, grounded in decision trees, is renowned for its prowess in enhancing predictive accuracy and promoting model generalization. Research has shown that this algorithm can reach an 87% accuracy[6]. The initial leg of our exploration involved the deployment of a standalone Decision Tree. We aimed to unravel patterns embedded in features like 'client,' 'datetime,' and 'user_agent,' all converging towards the classification of the 'Bot' target variable. However, as we endeavored to broaden our analytical scope, encompassing a more extensive dataset gleaned from the zanbil.ir e-commerce website, we grappled with the inherent challenges of the Decision Tree. Its vulnerability to overfitting and its limited adaptability to the intricate nuances of diverse web server activities became apparent.

The Random Forest emerged organically as a logical progression, presenting a solution to the constraints encountered with an individual Decision Tree. This ensemble methodology entails the creation of numerous decision trees, each trained on a distinct subset of the dataset, followed by the aggregation of their predictive prowess. The diversity within this forest, stemming from different dataset subsets and random feature selections, fosters the development of a robust and adaptable model capable of deciphering the intricacies present in comprehensive logs. Post one-hot encoding for the transformation of categorical features, the Random Forest showcased a substantial leap in accuracy compared to its solitary counterpart, achieving a commendable 63%. Diving deeper into the mechanics of the Random Forest, its proficiency arises from the amalgamation of multiple decision trees that operate independently yet contribute collectively. The algorithm introduces an element of randomness during training, both in terms of the data samples and feature subsets employed for each tree. This intentional variability ensures that each tree offers a unique perspective, fortifying the model's resilience to overfitting and enhancing its capability to capture diverse patterns within the dataset. The resultant ensemble effect facilitates a more nuanced and accurate bot detection mechanism, showcasing Random Forest's prowess in navigating the dynamic and diverse landscape of web server logs. This research trajectory not only underscored the advantages of ensemble learning but also highlighted the strategic amalgamation of algorithms as a potent strategy for effective bot detection within the dynamic and diverse web server log environment.

3.3. K-Nearest Neighbor

In our pursuit of refining bot detection strategies within web server logs, we turned our attention to the KNN algorithm. KNN is recognized for its straightforward approach, relying on the idea that similar data points in the feature space likely belong to the same class. KNN has been shown to be proficient at bot classification[10]

Understanding how KNN operates involves calculating distances between data points and their neighbors during training. The class assignment for a data point is determined by the majority class among its k-nearest neighbors. This proximity-based decision-making process makes KNN sensitive to the local structure of the data. In our practical application, after encoding features and working with a subset of the zanbil.ir e-commerce dataset, the KNN algorithm exhibited an observable accuracy rate of 46%.

Unlike the Decision tree and Random Forest tree, the KNN did not work well when operating with a small portion of the data. Additionally, as the load increased KNN produced better results. This would suggest that there were issues with the choices of k-values, or our distance formula.

3.4. Results

In our research, the Random Forest Tree (RFT) stood out as the most effective algorithm for bot detection, surpassing both the Decision Tree and k-Nearest Neighbors models. The RFT's notable success can be credited to its strength in overcoming the challenges faced by the standalone Decision Tree and KNN. While the Decision Tree worked well with smaller datasets, it struggled to maintain accuracy when dealing with the complexities of a larger dataset. On the other hand, KNN, recognized for its simplicity, might not have performed optimally due to poor choices in selecting k values. The RFT, as an ensemble method, effectively addressed overfitting issues associated with individual Decision Trees and exhibited adaptability to the nature of extensive web server logs. By aggregating insights from multiple decision trees and introducing randomness during training, the RFT captured subtle patterns within the data, leading to superior bot detection in the dynamic landscape of web server logs.

Data size	DT	RFT	KNN
1%	100%	100%	44%
5%	98%	100%	48%
Full data	40%	63%	46%

3.5. Future Work

In the future, our focus would be on improving the way we clean and handle the data. Better data quality is crucial for the success of machine learning models, so we had to pay more attention to details like handling missing values, dealing with outliers, and ensuring consistency in the dataset.

Additionally, we had to work on refining the heuristics for the 'is bot' attribute. Instead of sticking to basic transformations, like one-hot encoding, we had to dive deeper into feature engineering. For instance, we might explore patterns in the timing and frequency of user interactions to detect anomalies. Additionally, a more in-depth analysis of user-agent strings could provide valuable insights into distinguishing between bots and genuine users. These more advanced heuristics would give the models a more nuanced understanding of bot behavior.

For the k-Nearest Neighbors algorithm, we had taken a different approach. Trying out alternative distance formulas, such as Manhattan distance or Minkowski distance as opposed to Euclidean, could function better on the dataset. Also, we had to pay careful attention to choosing the right k values, conducting a thorough search for the optimal k, and considering factors like the dataset's complexity and dimensionality. This iterative improvement in both data preprocessing and algorithmic parameters would contribute to a more reliable and accurate bot detection system.

4. Conclusion

In the ever-changing world of online commerce, technology has ushered in unprecedented conveniences alongside a significant challenge – the intrusion of automated bots. This research sets out to bolster our defenses against these digital adversaries, leveraging the capabilities of machine learning algorithms, specifically focusing on decision trees, random forests, and KNN. Our exploration begins with a deep dive into a robust dataset sourced from zanbil.ir, an Iranian e-commerce site, offering a rich repository of web server logs that intricately detail user behaviors and server activities.

The decision trees provided us with a low accuracy, displaying results of around 40%. Working with a comprehensive dataset, the limitations of decision trees in handling intricacies became evident, prompting our shift towards ensemble methods like random forests. This transition led to a significant accuracy boost, demonstrating the effectiveness of ensemble learning in overcoming overfitting and capturing subtle patterns within diverse logs, resulting in a 63% accuracy.

Shifting focus to the KNN algorithm, we embraced its simplicity and proximity-based decision-making to uncover patterns in user interactions. While KNN showcased versatility with a 46% accuracy rate, its performance underscored the importance of thoughtful parameter choices. This comparative analysis, rooted in a nuanced understanding of the dataset, contributes valuable insights for practitioners navigating the landscape of bot detection.

In essence, this research not only sheds light on the strengths and limitations of decision trees, random forests, and KNN in the context of bot detection but also emphasizes the crucial role of dataset curation, feature engineering, and algorithmic diversity. By marrying human insights with machine learning techniques, our goal is to fortify e-commerce infrastructures, fostering an environment where fair competition thrives, and users confidently navigate the digital marketplace. As the e-commerce landscape evolves, the strategic integration of algorithms emerges as a powerful approach for effective bot detection within the dynamic web server log environment. The paper underscores the importance of combining human insight with machine learning techniques, offers a sophisticated defense against the persistent intrusions of bots in web server logs.

References:

[1] Xu, H. *et al.* (2018). Detecting and Characterizing Web Bot Traffic in a Large E-commerce Marketplace. In: Lopez, J., Zhou, J., Soriano, M. (eds) Computer Security. ESORICS 2018. Lecture Notes in Computer Science, vol 11099. Springer, Cham. https://doi.org/10.1007/978-3-319-98989-1_8

[2] G. Suchacka, "Analysis of aggregated bot and human traffic on e-commerce site," 2014 Federated Conference on Computer Science and Information Systems, Warsaw, Poland, 2014, pp. 1123-1130, doi: 10.15439/2014F346. keywords: {Robots;Crawlers;Web servers;Web pages;Search engines;Histograms},

[3] Hemmatpour, M., Zheng, C., & Zilberman, N. (2024). E-commerce bot traffic: in-network impact, detection, and mitigation.

[4] Kaushik, Dushyant and Gupta, Ankur and Gupta, Swati, E-Commerce Security Challenges: A Review (May 7, 2020). Proceedings of the International Conference on Innovative Computing & Communications (ICICC) 2020, Available at

SSRN: https://ssrn.com/abstract=3595304 or http://dx.doi.org/10.2139/ssrn.3595304

[5] Xu, H., Li, Z., Chu, C., Chen, Y., Yang, Y., Lu, H., Wang, H., & Stavrou, A. (2018). Detecting and characterizing web bot traffic in a large e-commerce marketplace. Computer Security, 143–163. https://doi.org/10.1007/978-3-319-98989-1_8

[6] Battur, R., & Yaligar, N. (July, 2019). Twitter Bot Detection using Machine Learning Algorithms. International Journal of Science and Research (IJSR), ISSN: 2319-7064.

[7] Baumgarten, R., Colton, S., & Morris, M. (2008). Combining AI methods for learning bots in a real-time strategy game. International Journal of Computer Games Technology, 2009, 1–10. https://doi.org/10.1155/2009/129075

[8] Wang X, Zheng Q, Zheng K, Sui Y, Cao S, Shi Y. Detecting Social Media Bots with Variational AutoEncoder and k-Nearest Neighbor. Applied Sciences. 2021; 11(12):5482. https://doi.org/10.3390/app11125482

[9] Hemmatpour, M., Zheng, C., & Zilberman, N. (2024). E-commerce bot traffic: in-network impact, detection, and mitigation.

[10] Ramalingaiah, A., Hussaini, S., & Chaudhari, S. (2021). Twitter bot detection using supervised machine learning. Journal of Physics: Conference Series, 1950(1), 012006. https://doi.org/10.1088/1742-6596/1950/1/012006