Performance Analysis of Machine Learning based Botnet Detection and Classification Models for Information Security

Salah-ddine KRIT[1]

[1] Professor of Computer Science, Ibn Zohr University, Agadir, Morocco

Emails: Salahddine.krit@gmail.com

Abstract

Botnet detection becomes a challenging issue in several domains like cybersecurity, finance, healthcare, law, order, etc. The botnet represents a set of cooperated Internet-linked devices managed by cyber criminals to start coordinated attacks and carry out different malicious events. As the botnets are seamlessly dynamic with the developing countermeasures presented by network and host-based detection schemes, conventional methods have failed to achieve enough safety for botnet threats. Therefore, machine learning (ML) models have been developed to detect and classify botnets for cybersecurity. In this view, this paper performs a comprehensive evaluation of different ML-based botnet detection and classification models. The botnet detection model involves a three-stage process, namely preprocessing, feature extraction, and classification. In this study, four ML models such as C4.5 Decision Tree, bagging, boosting, and Adaboost are employed for classification purposes. To highlight the performance of the four ML models, an extensive set of simulations was performed. The obtained results pointed out that the ML models can attain enhanced botnet detection performance.

Keywords: Information security, Botnet detection, Machine learning, Classification, Cybersecurity.

1. Introduction

The botnet is determined by the number of devices/computers dealt with the Internet for performing an unintentional malicious activity with no authentication of the data owners. Because of the continually developing behaviors of botnet, the traditional method fails to identify bonet [1]. The behavior modeling of botnets activity based on network connection was helpful to discover ambiguous botnet variants [2]. Botnet behaviors are modeled by determining the standard and frequent network activity as a pattern that is utilized for the recognition purpose. E.g., each bot must intermittently interconnect to the command and control (C&C) servers to receive orders or upgrade their status. The fundamental pattern of transmission activity denotes these transmission tracks for discovering botnet traffics [3]. In the previous years, malware (for example, malicious software) has become an increasing global consideration for countries, organizations, and users. Especially, compromising a group of infected hosts (also known as bots/zombies) is a severe threat to Internet privacy. A higher amount of fraudulent and malicious actions are performed in a shared way through this bot [4]. Therefore, computer systems have cooperated as botnet subjects to several attacks involving however unlimited to viruses, worms, social engineering, vulnerabilities, etc. They are handled with a C&C server that hides cybercriminals when executing their distributed and automated malicious challenges [5]. Fig. 1 illustrates the lifecycle of Botnet.

DOI: https://doi.org/10.54216/JCIM.000104

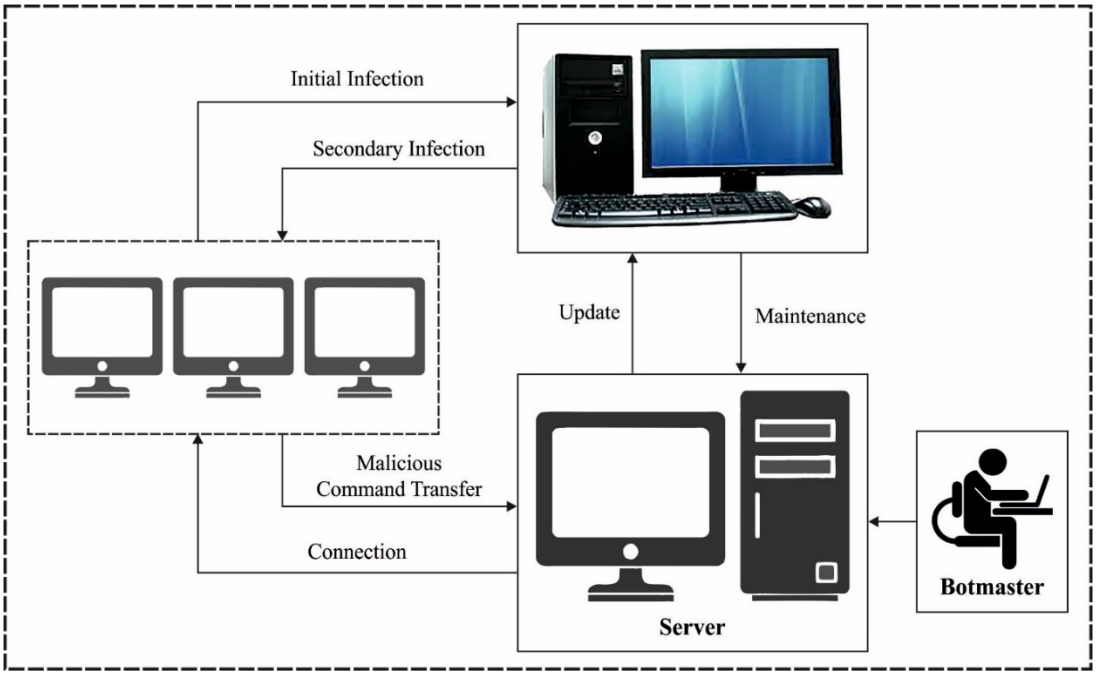

Fig. 1. Typical of Botnet cycle

Botnets could considerably harm the security of businesses and individuals. They pose severe and rising threats against cybersecurity since they offer a dispersed framework for several cybercrimes like DDoS attacks against malware dissemination, serious target, click fraud, and phishing [6]. Most of the present botnet recognition methods focus mainly on a certain botnet C&C protocol (for example, IRC, HTTP, etc.) and frameworks (P2P/centralized). They follow rule-based approach for detecting botnet in the networks [7]. But these methods could turn out to be obsolete and ineffective when botnet changes its framework and C&C protocols for evading detections [8]. Therefore, a strong botnet recognition method that might identify botnets with distinct features is of maximum significance. Beforehand examining the present botnet recognition system in this work, few research was initially made on anomaly recognition [9].

This paper performs a comprehensive evaluation of different ML-based botnet detection and classification models. The botnet detection model involves a three-stage process, namely preprocessing, feature extraction, and classification. In this study, four ML models such as C4.5 Decision Tree, bagging, boosting, and Adaboost are employed for classification purposes. To highlight the performance of the four ML models, an extensive set of simulations was performed. The obtained results pointed out that the ML models can attain enhanced botnet detection performance.

2. Prior Botnet Detection Approaches

Venkatachalam and Anitha [10] proposed a new host-based model to detect and differentiate Stegobot profile. The presented model also illustrates the capacity for detecting Stegobot network traffics, i.e., fundamentally distinct from legitimate multi-media social networks. The better performances of this method are proved on different social media datasets with particular estimation matrices. Several factors of multi-media features presented in this work assist in exploring the hidden transmission model of the botnets. Stegobot's profile stimulates genuine users and compromises another susceptible user in social networks.

Chowdhury et al. [11] presented new botnet detection methods based on topological features of the node with graphs: out-degree, in-degree, out-degree weight, in-degree weight, node betweenness, eigenvector centrality, and clustering coefficient. According to this feature, self-organizing map clustering methods are employed to establish a cluster of nodes in the network. This technique can isolate bots in clusters of smaller size when having the common node in similar big clusters. Gadelrab et al. [12] determine a thorough method for developing a botnet recognition model with ML methods. Recognizing botnet

DOI: https://doi.org/10.54216/JCIM.000104

member hosts or recognizing botnet traffics are the primary concept of several studies. This study aims to conquer two severe limits of present botnet recognition schemes: initially, the requirement for DPI and the requirement to gather traffic from various infected hosts. For achieving that, they examined many botnets samples of a recognized botnet.

Dorri et al. [13] address this shortcoming by providing SocialBotHunter, a semi-supervised collective classification technique that integrates the basic data of the social graphs on the user's social behaviors in a unified way for detecting social botnet in a Twitters as SNS. The result shows that SocialBotHunter can precisely detect social bot included in distributed social spam, called social spambot, with a lower FPR and satisfactory recognition time. Pektaş and Acarman [14] employ DNN for detecting botnets by modeling network traffic flows. Open source large-scale transmission tracks estimate the performances of the presented model. The simulation result shows that the DL method effectively identifies botnet traffic with higher TPs (attacks recognition accuracy) and lower FPR accuracy.

3. Proposed Botnet Detection Model

In this study, a botnet detection model is developed, which involves a three-stage process: preprocessing, feature extraction, and classification. In this study, four ML models such as C4.5 Decision Tree, bagging, boosting, and Adaboost are employed for classification purposes.

3.1 Preprocessing

To construct a bot detection architecture, they should generate specific social profile objects with the social networks. The object structure of profiles is a scheme have increasingly used feature of social media user accounts. A web crawler with distinct social networks API is used for collecting actual information dataset from open source data in OMSN user account. This technique might contain difficulties because of its massive amount of information, even though we considerably enhance the time required to classify profile objects as legitimate/infested profiles.

3.2 Feature Extraction

The attribute has single features of the user behaviors and profiles. Bot and genuine profiles consist of specific kinds of patter. E.g., a genuine user contains several legitimate friends, whereas bot is a fake profile and not once a response to the comment. The bot profile might contain several celebrity photos and attractive images and attracts another user for accepting yourself as a friend by initializing image sharing/friend request. The collection of attributes for detecting Bot is depended on the succeeding assumptions. Some infected profiles have an equivalent possibility of sending requests or infecting some vulnerable profiles in social media. The susceptible profile is infected once it has a relationship with or follows the infected bot profiles. It is appropriate for investigating numerous social graphs, profile-based features, and vulnerable image contents. The extracted feature depends on steganography, graph theory concepts, fundamental social networks, and their principles. They used image feature and user profile feature only for detecting bots. However, that could not recognize Bot communication. Therefore, social graph feature focuses on interactions among profiles are integrated for forming the feature sets. The social networks attribute the social graph-based relationship determined among profiles through image respond interactions, i.e., many potentials in online mult-imedia social networks. The concept is that this feature may take specific communication patterns, which might assist in distinguishing bot profile and genuine profile. The succeeding profile feature is extracted from an image respond user graphs that capture the number of interactions of the respective profile. The feature includes betweenness, cluster coefficient, assortativity, page rank, and reciprocity.

3.3 Classification Models

In this study, four ML models such as C4.5 Decision Tree, bagging, boosting, and Adaboost are employed for classification purposes.

DOI: https://doi.org/10.54216/JCIM.000104

3.3.1. C4.5 Decision Tree Model

The C4.5 DT technique tests the trained instances which have a similar outcome have been removed as it could not primary importance. So, it could not be limited from the DT when it does not have minimal two results with fewer samples. The candidate separates been occupied as to concern during the case which it could cut a particular amount of samples. There is an MDL based adjustment to separates numeric attributes. Quinlan planned a heuristic for avoiding over-fit. Afterward subtraction, it could discover that the information gain (IG) has been negative. When it does not have positive IG attributes that have been a type of prepruning, the tree stops growing. It represented as it could be unexpected for getting a pruned tree while postpruning is not active [15].

C4.5 utilizes the gain ratio:

$$P(D; D_1, \dots, D_k) = G(D; D_1, \dots, D_k) \cdot \left(-\sum_{i=1}^{k} \frac{|D_k|}{D} \log \frac{|D_k|}{D} \right)^{-1}, \quad (1)$$

which is different from the IG condition, taking normalized on the amount of feature values. The feature with maximum gain ratio, superior to average IGs is chosen as the splits.

3.3.2. Bagging Technique

The decision adopted in distinct learners is joined as to one forecast only. Joining individual's decision during the case of classifier has been voting. This technique has been utilized as combined as bagging as well as boosting. But, the individual techniques are resultant with bagging as well as boosting from distinct manners. Similar weights have been taken as the techniques from bagging, but weighting has been provided to further effective manners from growing as an executive could put alternative outcomes on the variation of expert's advice relying on its preceding correct evaluations [16]. The experts are individual DTs that are develop united as leads their vote on all tests. In this case, that one obtains further votes than the other class, it can be assumed as correct. If forecasts are developed as additional votes, it could be more reliable as different voters.

3.3.3. Boosting Technique

The boosting has been utilized to join several techniques for utilizing this idea as an attempt to find techniques that complete together. It can be the same as bagging in that it exploits voting to the resolves for classifying or to average the numeric evaluation to single individual techniques output. Another similarity has been that it takes composed manners, which are similar kind like DT. However, it can be iterative. But bagging generates utilize of individual manners that are developed individually, boosting utilizes novel techniques controlled with the efficiency of techniques developed before. The boosting reinforces novel techniques towards it become expert to samples contained in the wrong manner with earlier ones. Eventually, these techniques' influence has been evaluated as boosting their confidence, not permitting equivalent weight for every technique [17].

3.3.4. AdaBoost Technique

The AdaBoost has been a learning technique that considers a sample's weight as positive numbers. The existence of sample weights dependent upon an error of classification has been measured. The entire weights of the wrong classification samples can be divided as the whole weight of every sample, rather than the fraction of wrongly classified samples. If it can be weight samples, it could push the learning technique for focusing on a particular set of samples that are huge weight. There is great significance on such samples as it can be important for classifying them suitably. Every instance during the trained data allocated an equivalent weight with the boosting technique. Afterward, all samples are reweighted concerning the classifier outcome with the learning technique for creating the classifier. The weight of misclassified samples is improved, and that of the appropriately classified samples is decreased. Therefore, the simple instance has minimum weight, and the hard instance has maximum weight. The classifier has been generated from every following iteration to the reweighted data, which utilizes correctly categorize the hard instance. According to the outcome of this novel classifier, the weight of the sample has been decreased or improved. Therefore, but hard instances can develop harder, simple

DOI: https://doi.org/10.54216/JCIM.000104

instances can develop easier. But, few hard ones can develop easier and conversely [18]. All of these can be observed in one try. The weight often illustrates the sample has been wrongly classified with the classifier develop till now. At all samples, it can be the method of hardness measurement, which offers us an optimum manner of creating experts that complete together. The boosting has been superior to bagging from generating a classifier that implements optimum on novel data. Conversely, sometimes it fails from practical conditions with creating a classifier which is lesser success percentage than individual classifiers recognized in similar databases. AdaBoost represents that the joined classifiers are entirely equivalent to data.

4. Results and Discussion

The performance of the bot detection models takes place against four benchmark datasets, namely Facebook, Flickr, Barracuda labs, and ICWSM - 2013 dataset.

Table 1 and Fig. 2 illustrate the performance analysis of the applied models on the Facebook dataset. The results have shown that the SVM model has showcased poor outcomes over the other ML models. The Boosting and K-NN techniques have obtained slightly enhanced performance, whereas the Bagging and DT models have gained moderately closer outcomes. Moreover, the C4.5 and NB models have demonstrated reasonable outcomes. However, the AdaBoost model has accomplished effective outcomes with the higher TP, TN, FP, FN, and accuracy of 0.9725, 1.0000, 0.0000, 0.0275, and 0.9734, respectively.

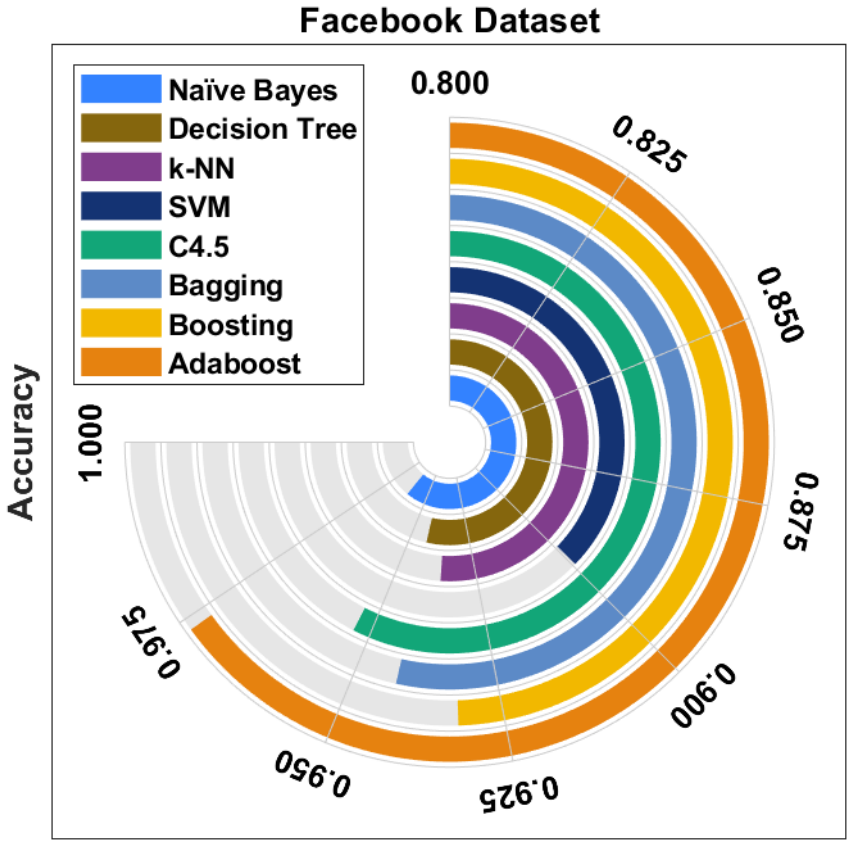

Fig. 2. Result analysis of detection models on Facebook Dataset

DOI: https://doi.org/10.54216/JCIM.000104

Table 1 Result analysis of detection models on Facebook Dataset

Classifiers	TP	TN	FP	FN	Accuracy
Naïve Bayes	0.9613	1.0000	0.0000	0.0387	0.9624
Decision Tree	0.9228	0.9751	0.0249	0.0772	0.9430
k-NN	0.9315	0.9854	0.0146	0.0685	0.9361
SVM	0.9042	0.9783	0.0217	0.0958	0.8993
C4.5	0.9424	0.9931	0.0069	0.0576	0.9534
Bagging	0.9302	0.9975	0.0025	0.0698	0.9426
Boosting	0.9129	0.9977	0.0023	0.0871	0.9320
Adaboost	0.9725	1.0000	0.0000	0.0275	0.9734

Table 2 Result analysis of detection models on Flickr Dataset

Classifiers	TP	TN	FP	FN	Accuracy
Naïve Bayes	0.9623	1.0000	0.0000	0.0377	0.9626
Decision Tree	0.9437	0.9589	0.0444	0.0563	0.9498
k-NN	0.9236	0.9642	0.0358	0.0765	0.9368
SVM	0.8964	1.0000	0.0000	0.1036	0.8993
C4.5	0.9162	0.9951	0.0049	0.0838	0.9263
Bagging	0.9288	0.9724	0.0276	0.0712	0.9356
Boosting	0.9099	0.9827	0.0173	0.0901	0.9156
Adaboost	0.9738	1.0000	0.0000	0.0262	0.9772

Table 2 and Fig. 3 showcases the performance analysis of the applied techniques on the Flickr dataset. The outcomes exhibited that the SVM approach has illustrated worse outcomes over the other ML techniques. Also, the Boosting and K-NN systems have gained somewhat improved performance, whereas the Bagging and DT approaches have reached moderately closer outcomes. In addition, the C4.5 and NB approaches have outperformed reasonable outcomes. Finally, the AdaBoost approach has accomplished effectual results with superior TP, TN, FP, FN, and accuracy of 0.9738, 1.0000, 0.0000, 0.0262, and 0.9772.

DOI: https://doi.org/10.54216/JCIM.000104

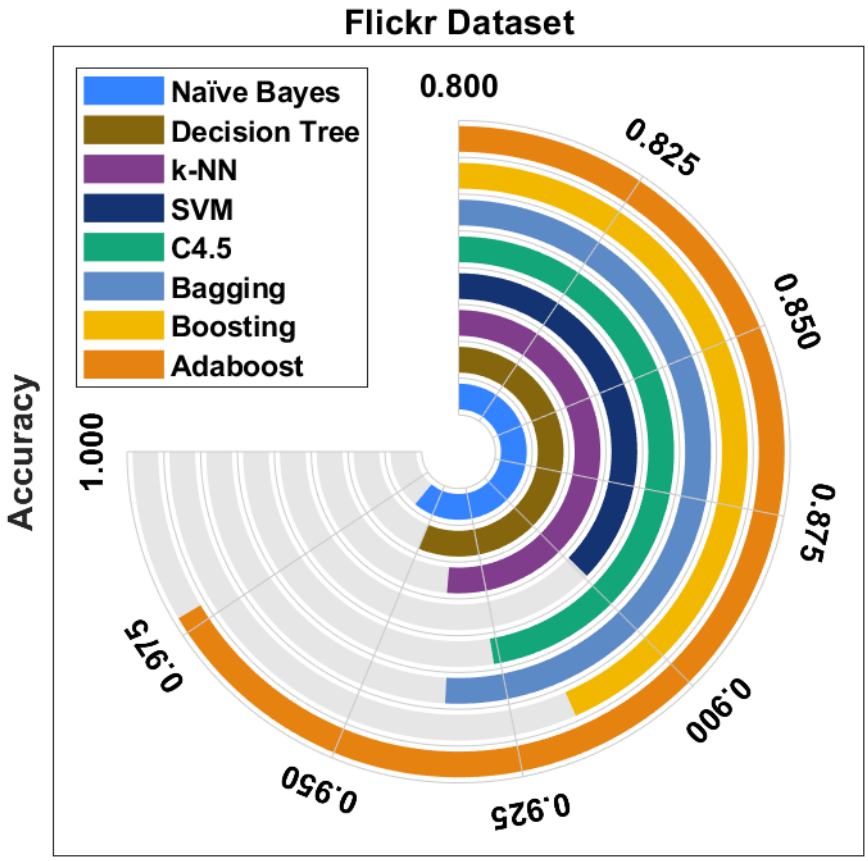

Fig. 3. Result analysis of detection models on Flickr Dataset

Table 3 Result analysis of detection models on Barracuda labs Dataset

Classifiers	TP	TN	FP	FN	Accuracy
Naïve Bayes	0.9544	0.9876	0.0124	0.0456	0.9224
Decision Tree	0.9455	0.9876	0.0124	0.0545	0.9230
k-NN	0.9268	0.9875	0.0125	0.0732	0.9610
SVM	0.8840	0.9884	0.0116	0.1160	0.8712
C4.5	0.9406	0.9912	0.0088	0.0594	0.9512
Bagging	0.9483	0.9893	0.0107	0.0597	0.9554
Boosting	0.9598	0.9886	0.0114	0.0502	0.9620
Adaboost	0.9658	0.9921	0.0079	0.0342	0.9821

Table 3 and Fig. 4 demonstrate the performance analysis of the applied approaches on the Barracuda labs dataset. The results depicted that the SVM system has been shown lower outcomes over the other ML models. Similarly, the Boosting and K-NN schemes have attained somewhat higher performance, whereas the Bagging and DT approaches have gained moderately closer outcomes. Moreover, the C4.5 and NB models have demonstrated reasonable outcomes. At last, the AdaBoost model has accomplished effective outcomes with the maximum TP, TN, FP, FN, and accuracy of 0.9658, 0.9921, 0.0079, 0.0342, and 0.9821 respectively.

DOI: https://doi.org/10.54216/JCIM.000104

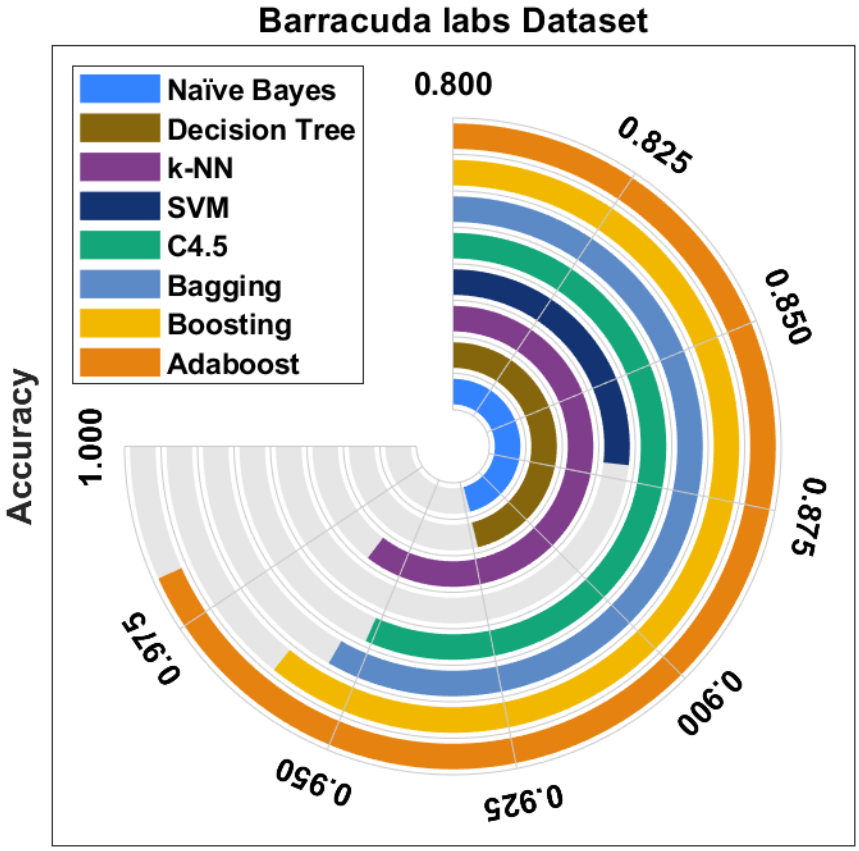

Fig. 4. Result analysis of detection models on Barracuda labs Dataset

Table 4 Result analysis of detection models on ICWSM - 2013 Dataset

Classifiers	TP	TN	FP	FN	Accuracy
Naïve Bayes	0.9500	0.9893	0.0107	0.0500	0.9626
Decision Tree	0.9453	0.9921	0.0079	0.0547	0.9790
k-NN	0.9192	0.9905	0.0095	0.0808	0.9790
SVM	0.9610	0.8712	0.0059	0.1208	0.9630
C4.5	0.9319	0.9728	0.0272	0.0681	0.9485
Bagging	0.9515	0.9906	0.0094	0.0485	0.9654
Boosting	0.9595	0.8988	0.1012	0.0405	0.9678
Adaboost	0.9723	0.9947	0.0053	0.0277	0.9854

Table 4 and Fig. 5 depict the performance analysis of the applied techniques on the ICWSM - 2013 dataset. The outcomes exhibited that the SVM methodology has illustrated the least outcome over the other ML manners. Besides, the Boosting and K-NN algorithms have gained slightly enhanced performance, whereas the Bagging and DT models have gained moderately closer outcomes. Moreover, the C4.5 and NB models have outperformed reasonable outcomes. At last, the AdaBoost method has accomplished effectual outcomes with the higher TP, TN, FP, FN, and accuracy of 0.9723, 0.9947, 0.0053, 0.0277, and 0.9854 correspondingly.

DOI: https://doi.org/10.54216/JCIM.000104

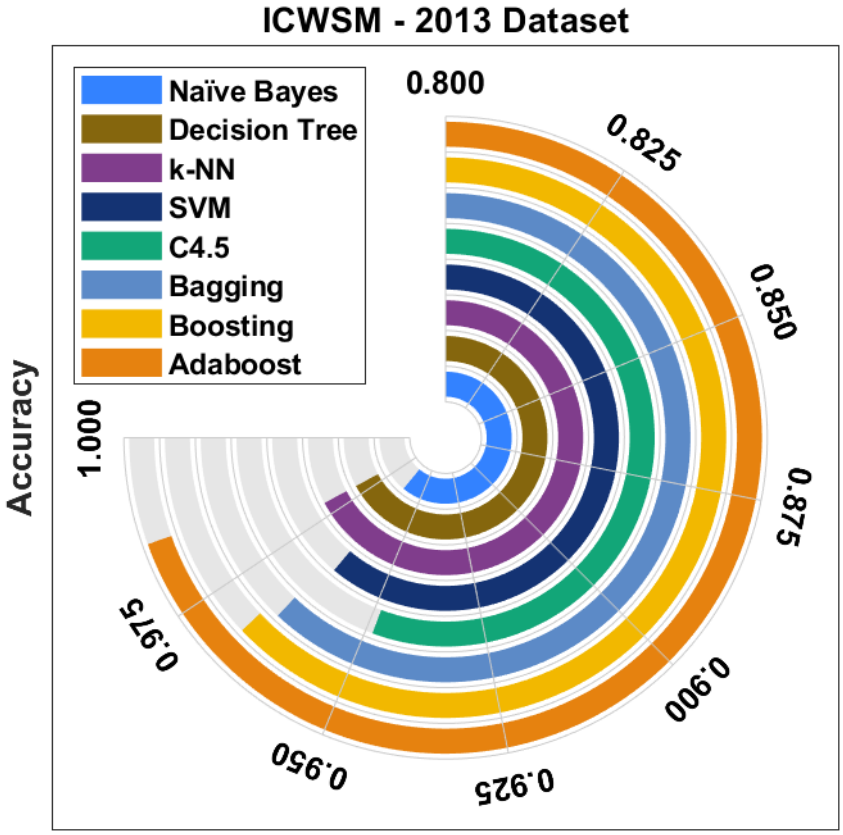

Fig. 5. Result analysis of detection models on ICWSM - 2013 Dataset

5. Conclusion

In this paper, the botnet detection model is presented to attain information security. The botnet detection model involves a three-stage process, namely preprocessing, feature extraction, and classification. In this study, four ML models such as C4.5 Decision Tree, bagging, boosting, and Adaboost are employed for classification purposes. To highlight the performance of the four ML models, an extensive set of simulations was performed. The obtained results pointed out that the ML models have the ability to attain enhanced botnet detection performance. As a part of the future scope, deep learning (DL) models can be developed to boost botnet detection efficiency.

References

[1] Karim, A., Salleh, R.B., Shiraz, M., Shah, S.A.A., Awan, I. and Anuar, N.B., 2014. Botnet detection techniques: review, future trends, and issues. Journal of Zhejiang University SCIENCE C, 15(11), pp.943-983.

[2] Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A. and Garant, D., 2013. Botnet detection based on traffic behavior analysis and flow intervals. computers & security, 39, pp.2-16.

[3] Barford P, Yegneswaran V. An inside look at botnets. Special workshop on malware detection: advances in information security; 2006.

[4] Zeidanloo HR, Shooshtari MJZ, Amoli PV, Safari M, Zamani M. A taxonomy of botnet detection techniques. In: 2010 3rd IEEE international conference on computer science and information technology (ICCSIT), vol 2. New York: IEEE;. 2010. p. 158–62.

[5] Sonawane SR. A review on botnet and botnet detection methods. Int J Comput Sci Innov. 2016;1:107–16.

[6] Zhang J, Perdisci R, Lee W, Sarfraz U, Luo X. Detecting stealthy P2P botnets using statistical trafc fngerprints. In: 2011 IEEE/IFIP 41st international conference on dependable systems & networks (DSN). New York: IEEE; 2011. p. 121–32.

DOI: https://doi.org/10.54216/JCIM.000104

[7] Zeidanloo, H.R., Shooshtari, M.J.Z., Amoli, P.V., Safari, M. and Zamani, M., 2010, July. A taxonomy of botnet detection techniques. In 2010 3rd International Conference on Computer Science and Information Technology (Vol. 2, pp. 158-162). IEEE.

[8] Zhao, Y., Xie, Y., Yu, F., Ke, Q., Yu, Y., Chen, Y. and Gillum, E., 2009, April. BotGraph: Large Scale Spamming Botnet Detection. In NSDI (Vol. 9, pp. 321-334).

[9] Gu, G., Perdisci, R., Zhang, J. and Lee, W., 2008. Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection.

[10] Venkatachalam, N. and Anitha, R., 2017. A multi-feature approach to detect Stegobot: a covert multimedia social network botnet. Multimedia Tools and Applications, 76(4), pp.6079-6096.

[11] Chowdhury, S., Khanzadeh, M., Akula, R., Zhang, F., Zhang, S., Medal, H., Marufuzzaman, M. and Bian, L., 2017. Botnet detection using graph-based feature clustering. Journal of Big Data, 4(1), pp.1-23.

[12] Gadelrab, M.S., ElSheikh, M., Ghoneim, M.A. and Rashwan, M., 2018. BotCap: Machine learning approach for botnet detection based on statistical features. Int. J. Commun. Netw. Inf. Secur, 10(3), p.563.

[13] Dorri, A., Abadi, M. and Dadfarnia, M., 2018, August. SocialBotHunter: Botnet detection in Twitter-like social networking services using semi-supervised collective classification. In 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech) (pp. 496-503). IEEE.

[14] Pektaş, A. and Acarman, T., 2018. Botnet detection based on network flow summary and deep learning. International Journal of Network Management, 28(6), p.e2039.

[15] Dai, W. and Ji, W., 2014. A mapreduce implementation of C4. 5 decision tree algorithm. International journal of database theory and application, 7(1), pp.49-60.

[16] Chi, M. and Bruzzone, L., 2005. A semilabeled-sample-driven bagging technique for ill-posed classification problems. IEEE Geoscience and Remote Sensing Letters, 2(1), pp.69-73.

[17] Mesgarani, A., Alam, M.N., Nelson, F.Z. and Ay, S.U., 2010, August. Supply boosting technique for designing very low-voltage mixed-signal circuits in standard CMOS. In 2010 53rd IEEE international midwest symposium on circuits and systems (pp. 893-896). IEEE.

[18] Yuan, Y., Kaklamanos, G. and Hogrefe, D., 2016, November. A novel semi-supervised adaboost technique for network anomaly detection. In Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (pp. 111-114).

DOI: https://doi.org/10.54216/JCIM.000104