iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.68351

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Detecting Botnet Activity: Learning Discriminative Boosted Bayesian Networks for Accurate Analysis

Vaishnavi Cherukuvada[1], Arun Kumar Tomar[2], Tatapudi Aishwarya Anand[3], Aditya Poddar[4]
[1]*IT, Accenture, Hyderabad, India*
[2]*Computer Science, Rajkiya Engineering College, Sonbhadra, Uttar Pradesh, India*
[3]*IT, Tata Consultancy Services, Telangana, India*
[4]*IT, Spikewell, Odisha, India*

Abstract: ***Distributed network attacks, including botnets, pose significant chal- lenges in detecting and mitigating their activities. We present the applicationof learning Discriminative Boosted Bayesian Networks to detect botnet activ-ity using the CTU-13-Dataset. Our results are compared with traditional machine learning approaches, with and without expert knowledge. This marks the first application of statistical relational learning in this domain, addressing the need for effective detection in evolving threat landscapes. Our approach focuses on learning a generalized model from sparse botnet data, addressing the challenges of limited data availability. By carefully engineering features and selecting ap- propriate learning algorithms, we aim to achieve accurate results. The CTU-13- Dataset, capturing diverse botnet examples, is utilized for experiments. Our re- search contributes to intrusion detection and botnet detection by emphasizing the importance of domain knowledge in feature engineering.***

Keywords: Botnet Detection, Machine Learning, Feature Engineering, Cybersecurity, Bayesian Networks

I. INTRODUCTION

Distributed network attacks have been a thorn in the internet since its early days. In addition to denial-of-service, they are responsible for spreading spam and malware andeven have a hand in data exfiltration and theft. Despite much effort in the research andwhite hat communities, these attacks are more prevalent today than ever.

Distributed network attacks are commonly spread by the use of *botnets*, which are a network of (often) hijacked computers to accomplish some nefarious goal, such as flooding a web server with page requests. While reducing the frequency of these attacksis ideal, the evolution of attack strategies in response to current detection and mitigation techniques requires an approach that can generalize to newer, potentially unobserved distributed attacks.

However, given the growth of distributed attacks within the past 10 years, how well do these approaches generalize to the current threat landscape? This question has im- plications for practically every aspect of the machine learning pipeline: data collection, feature extraction, which model to learn, and which algorithm to train the model.

We focus on the problem of learning a generalizable model from *sparse* botnet data, which often arises in the real world. This occurs typically due to the overall periodspent collecting network traffic compared to the duration the botnet was active. Despitethis sparsity, careful feature engineering should help prevent data overfitting and allow for some acceptable measure of accuracy for many different off-the-shelf learning algorithms. We hope carefully selecting our learning algorithm will enable model generality on related data.

Our experiments use the *CTU-13-Dataset*, which is "a dataset of botnet traffic cap-tured in the CTU University, Czech Republic, in 2011" [8]. This dataset comprises 13different runs utilizing multiple disjoint botnets. Most alluring to us is the sparsity of the botnet examples; a vast majority (>97%) of the examples are from expected trafficflows. Also of importance is the use of multiple botnets over the runs which make up the datasets. Such diversity is invaluable to experiment with generality across multiplesimilar botnets, in which some of the botnets may not be observed during training.

The rest of the paper is structured as follows: In the related work section, we discuss related work that is both relevant to the general field of intrusion detection and the subset of botnet detection. Also presented is work on why domain knowledge is essential in feature engineering. Next, we dive into statistical relational learning and related work relevant to our problem definition and approach. The experimental section showcases results from the feature engineering stage, and we wrap up the paper with a thorough discussion on inherent limitations that prevented full results from our chosen approachfrom reaching fruition.

©IJRASET: All Rights are Reserved

II. RELATED WORK

A. Traditional Botnet Detection

As the number of people and devices accessing the internet increases, the need to deal with problems such as intrusion detection, denial of service attacks, and botnets in- creases as well.

Many approaches have applied traditional machine learning techniques for botnet and general distributed denial-of-service classification tasks, often with good results on specific network datasets. Earlier work tended to focus on general intrusion detection tasks, often in a temporal environment [25, 16, 10, 22]. In this setting, intrusions were modeled as events ordered by time, with each set of events serving as a single example of either normal or abnormal behavior. Markov chains and hidden Markov networks are popular models to learn in this setting.

In a non-temporal setting, examples tend to be individual events, with each event corresponding to one or more features that describe the event. For network security- related intrusion tasks, these features are commonly those found in net flow data: source and destination IP address and port, number of packets comprising the flow, total bytes sent or received, direction of flow, etc. Early work in this setting trained models such as mixture models [17], Naive Bayes [4], Bayesian networks [11, 21], random forests [26], and decision trees [15].

Early work specifically on botnet detection focused more on the command and con- trol (c&c) flows [12, 9, 5, 3, 7], which is traffic generated by a botnet for node coordi- nation. A comparison of C4.5, naive Bayes, and Bayesian network learners to classify botnet c&c traffic from regular traffic was spotlighted in [12]. The Bayesian network results especially highlighted the issue of model overfitting to a given training set when confronted with test data that may have originated from the same botnet over different runs. We imagine this problem is compounded when factoring in test data from similar botnets.

More recent work has applied a random forest learner to a large-scale botnet dataset captured by UC San Diego [19]. Most of their work went into the feature engineer- ing and underlying distributed framework to handle the sheer amount of data present, but the results were promising. Feature engineering related explicitly to botnet c&c channels was discussed in [1], using C4.5 as the learning algorithm. Their approach to feature engineering used a genetic algorithm and exhaustive search to select essen- tial features from a list of non-temporal features generated by aggregating temporal flow data. Given the small feature space to search (19), and their results for features selected, we feel that domain knowledge can reproduce such a set of essential features. One drawback to much of the previous work in this field is how well a trained model represents reality. For example, the dataset used in [26] was released in 1999, almost a decade before the paper was published. The network and threat landscape changed considerably during that time. In [21], the period between the dataset they used and the publishing of their paper was 12 years. More recently, [1] used a dataset released in 2011.

Another factor to consider is the feature engineering itself. Plug-and-play methods may do well on a single dataset with minimal domain knowledge, but what happens when applying the trained model to other datasets showcasing the same class of threats? Is the generality there? How much domain knowledge is needed to maintain a baseline of generality? [2] took a deep look into this problem, and, perhaps unsurprisingly, do- main experts are pretty crucial during the feature engineering task. This suggests that a plug-and-play approach to botnet detection may not yield the best results, but rather, a combination of a domain expert and a 'model' expert might be preferable.

B. Statistical Relational Learning

The imbalance between the potential number of positive and negative examples makes this a potential problem to approach from the perspective of statistical relational learn- ing (SRL). The authors apply the state-of-the-art statistical relational learning system: BoostSRL[5] based on the relational functional gradient boosting algorithm [13]. Gradi- ent boosted tree learners generally set up the problem in the form of learning a series of regression trees, where each tree is a relatively weak learner that fits toward correcting the error of the previous.

BoostSRL (an implementation of the Relational Functional Gradient Boosting al- gorithm) has been applied to a variety of real-world domains; including identifying Parkinson's patients, predicting the onset of postpartum depression, and recommending jobs to potential applicants [6, 14, 24].

The authors build on the work of [18, 23] for learning discriminative-boosted Bayesian networks. Because statistical relational models may operate over data with a massive imbalance between the number of positive and negative examples and different costs for

[5] https://github.com/starling-lab/BoostSRL

Classification, explicitly tweaking the cost function for this problem is essential. Usu- ally, the cost function is tweaked explicitly to set the trade-off between false negativesand false positives– since in recommendation systems, the goal is to have high preci- sion, but in medical applications, high recall is more desirable–but explicitly deciding on the trade-off in the botnet problem is not as obvious. If the goal is to identify bots with high precision, some may not be marked as bots under cases of uncertainty; if highrecall is desired, some humans may also be labeled as bots. This point is explored by tweaking alpha and beta values in our experiments.

III. EXPERIMENTS

We answer four questions: (1) How do standard machine learning techniques perform on this task? (2) Do relational learning techniques–notably discriminative boosted Bayesian networks–provide better generalization? (3) Can we make general observations about alpha and beta values for tweaking the cost of false positives or negatives in practice? (4) Given the general knowledge, can we retroactively tweak the standard machine learn- ing techniques to produce superior results? (5) What is the value of expert knowledge in such a task?

A. Results

CTU-13-Dataset		
Algorithm	Training Ac- curacy	Testing Accuracy
BN CI	100.0%	0.0%
BN CI2	99.9%	0.1%
BN CI3	99.6%	0.4%
BN Tabu	100.0%	0.0%
BN Tabu2	99.9%	0.1%
BN Tabu3	99.5%	0.5%
Random Forest	100.0%	0.0%
Naive Bayes	95.5%	4.5%

Before diving into the model training, we need to remove features of our original set, which will probably not be generalized. Eight Weka [20] algorithms were used to perform these baseline experiments, using all the features in our set. Since the goal is to learn a general method for detecting botnet activity, each of the thirteen CTU "tasks" was treated as a fold, and we report the average accuracy when the classifier is trainedon one task and tested on another.

As reflected in these results, there are present features that are not generalizing to other runs within the dataset. Now, the task becomes one of removing the 'bad' featuresfrom the set. Determining such features can be accomplished through domain knowl- edge and analysis of the Bayesian network structures learned from the table above.

Regarding domain knowledge, we observe that features such as source IP addresses will naturally overfit to a training set due to the implicit assumption this feature makes that all botnets will originate from the same set of source IP addresses. This is not true, even for the same botnet. For example, botnets that utilize source IP spoofing, even the same botnet will appear to come from different sets of source IP addresses over different experimental runs. In analyzing the Bayesian network structures learned from the table above, we ob- serve that given the single-class tree-based networks learned, the children (feature) nodes have more influence to the class the closer they are to the class in terms of edges. So direct children will hold more influence than grandchildren. Indeed, the source IP address (SrcAddr) was a direct child in all the models learned. This doesn't necessar-ily mean the feature is bad, just that it will hold a lot of influence, so we need to look at all such features directed at children. One feature that falls into this boat, destinationIP address (DstAddr), should also be removed because a botnet may not always at- tack the same destination and will not generalize to data collected from other network topologies. Removing the destination IP address as a feature is an example of applyingdomain knowledge after using other techniques to point in a specific direction. One exciting feature that usually is a direct influence but which does not generalize at all is start time (StartTime). When a botnet begins its attack, relative to the start of collecting data, it is determined by the specific dataset run. Outside of the dataset, it's practically irrelevant but sometimes can give the illusion of being a good feature depending on exactly when the attack begins relative to whatever else is going on at that point in time. And that's the key; the probability of other events sharing the exact timestamp of the start of the attack is rare.

Other features that hold a direct influence on the class but which should generalize well are destination port (Dport) and flow duration (Dur). For the former, botnets typically will target a specific port on a victim server for a specific protocol (say HTTP), so the destination port should generalize to most botnets of the same attack class. The same argument can be used for keeping flow duration as well.

The next step is finding an appropriate ordering of the features we keep, which is a prerequisite for using discriminative-boosted Bayesian networks as a learning method. The binetflow in the CTU-13 Dataset supplies the following fourteen labels:

StartTime, Dur, Proto, SrcAddr, Sport, Dir, DstAddr, Dport, sTos, dTos, TotPkts, TotBytes, SrcBytes, Label Of these labels, the target is the value in the Label column, and after elimi- nating the features mentioned above, the variable ordering becomes: Dport, Dur, TotBytes, TotPkts, SrcBytes, Proto, Dir, Sport.

B. Limitations

The code is implemented, but we do not have the results for the thirteen tasks yet. Whena task is converted to the appropriate predicate-logic format, there are around 25 millionfacts and several million positive and negative examples. Despite how much computingpower is thrown at it, BoostSRL appears to hang while reading the facts and does not recover.

There are several possible ways around this. The variable ordering we are currentlyusing may be adapted to use even fewer variables, iteratively removing one variable (starting from the end deemed “least relevant”) until we have something that can be computed. A more efficient representation of the facts may be possible–currently, we have discretized the positive and negative examples into “bot or not”, but similar dis- cretizations may be possible for the facts may reduce the overall number of groundingsthat need to be reasoned about. If neither of these accomplishes our goals, a more robust learning and inference framework may need to be considered.

IV. CONCLUSION

While the experiments still may need considerable work, the authors have presented a novel approach toward detecting botnet activity on a network–as far as we know, this isthe first application of statistical relational learning to this domain.

V. ACKNOWLEDGEMENTS

The authors thank Professor Sriraam Natarajan, Professor Gautam Kunapuli, and their classmates in the Spring 2018 Statistical Relational Learning Seminar (CS 7301.002) for their comments and general feedback as this paper came together.

REFERENCES

[1] Alejandre, F.V., Corte´s, N.C.,Anaya, E.A.: Feature selection to detect botnets using machine learning algorithms. In: 2017 International Conference on Electronics, Communications and Computers (CONIELECOMP). pp. 1–7 (Feb 2017). https://doi.org/10.1109/CONIELECOMP.2017.7891834

[2] Ben-Asher, N., Gonzalez, C.: Effects of cyber security knowl- edge on attack detection. Computers in Human Behavior 48, 51-61 (2015). https://doi.org/https://doi.org/10.1016/j.chb.2015.01.039, http://www.sciencedirect.com/science/article/pii/S0747563215000539

[3] Bilge, L., Balzarotti, D., Robertson, W., Kirda, E., Kruegel, C.: DISCLO- SURE: Detecting botnet command and control servers through large-scale net- flow analysis. In: ACSAC 2012, 28th Annual Computer Security Applications Conference, December 3-7, 2012, Orlando, Florida, USA. Orlando, UNITED STATES (12 2012). https://doi.org/http://dx.doi.org/10.1145/2420950.2420969, http://www.eurecom.fr/publication/3886

[4] Bringas, P.G., Penya, Y.K.: Next-generation misuse and anomaly prevention system. In: Fil- ipe, J., Cordeiro, J. (eds.) Enterprise Information Systems. pp. 117–129. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)

[5] Cho, C.Y., Babi c´, D., Shin, E.C.R., Song, D.: Inference and analysis of formal models of botnet command and control protocols. In: Proceedings of the 17th ACM Conference on Computer and Communications Security. pp. 426–439. CCS ’10, ACM, New York, NY, USA (2010). https://doi.org/10.1145/1866307.1866355, http://doi.acm.org/10.1145/1866307.1866355

[6] Dhami, D.S., Soni, A., Page, D., Natarajan, S.: Identifying parkinson’s patients: A functional gradient boosting approach. In: Conference on Artificial Intelligence in Medicine in Europe. pp. 332–337. Springer (2017)

[7] Dietrich, C.J., Rossow, C., Pohlmann, N.: Cocospot: Clustering and recognizing botnet command and control channels using traffic analysis. Computer Networks 57(2), 475 – 486 (2013). https://doi.org/https://doi.org/10.1016/j.comnet.2012.06.019, http://www.sciencedirect.com/science/article/pii/S1389128612002472, botnet Activity: Analysis, Detection and Shutdown

[8] Garcia, S., Grill, M., Stiborek, J., Zunino, A.: An empirical comparison of botnet detection methods. Computers & Security **45**, 100–123 (2014)

[9] Gu, G., Zhang, J., Lee, W.: Botsniffer: Detecting botnet command and control channels in network traffic. In: NDSS (2008)

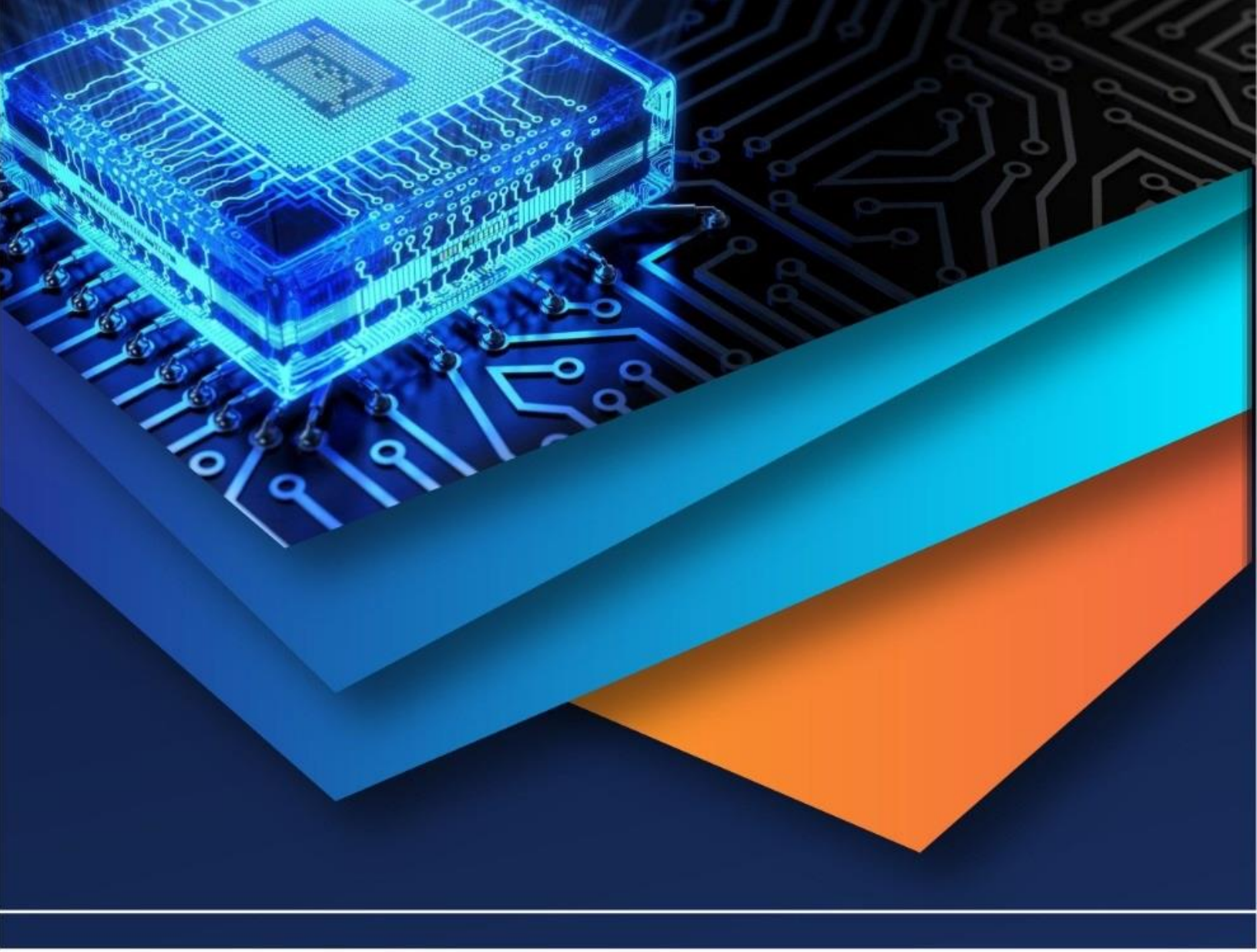
[10] Joshi, S.S., Phoha, V.V.: Investigating hidden markov models capabilities in anomaly detection. In: Proceedings of the 43rd Annual Southeast Regional Conference- Volume 1. pp. 98–103. ACM-SE 43, ACM, New York, NY, USA (2005). https://doi.org/10.1145/1167350.1167387, http://doi.acm.org/10.1145/1167350.1167387

[11] Kruegel, C., Mutz, D., Robertson, W., Valeur, F.: Bayesian event classification for intrusion detection. In: 19th Annual Computer Security Applications Conference, 2003. Proceedings.pp. 14–23 (Dec 2003). https://doi.org/10.1109/CSAC.2003.1254306
[12] Livadas, C., Walsh, R., Lapsley, D., Strayer, W.T.: Usilng machine learning technliques to identify botnet traffic. In: Proceedings. 2006 31st IEEE Conference on Local Computer Net-works. pp. 967–974 (Nov 2006). https://doi.org/10.1109/LCN.2006.322210
[13] Natarajan, S., Kersting, K., Khot, T., Shavlik, J.: Boosted statistical relational learners: From benchmarks to data-driven medicine. Springer (2015)
[14] Natarajan, S., Prabhakar, A., Ramanan, N., Bagilone, A., Siek, K., Connelly, K.: Boosting for postpartum depression prediction. In: Connected Health: Applications, Systems and En-gineering Technologies (CHASE), 2017 IEEE/ACM International Conference on. pp. 232– 240. IEEE (2017)
[15] Osanaiye, O., Cai, H., Choo, K.K.R., Dehghantanha, A., Xu, Z., Dlodlo, M.: Ensemble- based multi-filter feature selection method for ddos detection in cloud computing. EURASIP Journal on Wireless Communications and Networking **2016**(1), 130 (May 2016). https://doi.org/10.1186/s13638-016-0623-3, https://doi.org/10.1186/s13638-016-0623-3
[16] Ourston, D., Matzner, S., Stump, W., Hopkins, B.: Applications of hidden markov models to detecting multi-stage network attacks. In: 36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the. pp. 10 pp.– (Jan 2003). https://doi.org/10.1109/HICSS.2003.1174909
[17] Puttini, R.S., Marrakchi, Z., Mé, L.: Bayesian classification model for real-time intrusion detection. In: In 22th International Workshop on Bayesian Inference and Maximum EntropyMethods in Science and Engineering (2002)
[18] Ramanan, N., Yang, S., Grannis, S., Natarajan, S.: Discriminative boosted bayes networks for learning multiple cardiovascular procedures. In: 2017 IEEE International Conference onBioinformatics and Biomedicine (BIBM). pp. 870–873. IEEE (2017)
[19] Singh, K., Guntuku, S.C., Thakur, A., Hota, C.: Big data analytics framework for peer-to-peer botnet detection using random forests. Information Sciences**278**, 488 – 497 (2014).
https://doi.org/https://doi.org/10.1016/j.ins.2014.03.066, http://www.sciencedirect.com/science/article/pii/S0020025514003570
[20] Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann (2016
[21] Xu, J., Shelton, C.R.: Intrusion detection using continuous time bayesian networks. J. Artif. Int. Res. **39**(1), 745–774 (Sep 2010), http://dl.acm.org/citation.cfm?id=1946417.1946434
[22] Xu, X., Sun, Y., Huang, Z.: Defending ddos attacks using hidden markov models and co-operative reinforcement learning. In: Proceedings of the 2007 Pacific Asia Conference on Intelligence and Security Informatics. pp. 196–207. PAISI'07, Springer-Verlag, Berlin, Hei- delberg (2007), http://dl.acm.org/citation.cfm?id=1763599.1763621

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

1. Yang, S., Khot, T., Kersting, K., Kunapuli, G., Hauser, K., Natarajan, S.: Learning from imbalanced data in relational domains: A soft margin approach. In: Data Mining (ICDM), 2014 IEEE International Conference on. pp. 1085–1090. IEEE (2014)
2. Yang, S., Korayem, M., AlJadda, K., Grainger, T., Natarajan, S.: Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive statistical rela- tional learning approach. Knowledge-Based Systems **136**, 37–45 (2017)
3. Ye, N.: A markov chain model of temporal behavior for anomaly detection. In: In Pro- ceedings of the 2000 IEEE Workshop on Information Assurance and Security. pp. 171–174(2000)
4. Zhang, J., Zulkernine, M., Haque, A.: Random-forests-based net-work intrusion detection systems. Trans. Sys. Man Cyber Part C **38**(5), 649–659 (Sep 2008). https://doi.org/10.1109/TSMCC.2008.923876, http://dx.doi.org/10.1109/TSMCC.2008.923876

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)