International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 4| ISSN: 2456-3307

# Identifying Malicious Web Links and Their Attack Types in Social Networks

**R. Hamsa Veni[1], A.Hariprasad Reddy[2], C.Kesavulu[3]**

[1]Assistant Professor, Department of MCA, Sri Venkateswara College of Engineering and Technology, Chittoor, AP, India,

[23]PG Scholar, Department of MCA, Sri Venkateswara College of Engineering and Technology, Chittoor, AP, India

## ABSTRACT

Malicious URLs are wide wont to mount numerous cyber attacks together with spamming, phishing and malware. Detection of malicious URLs and identification of threat varieties area unit important to thwart these attacks. Knowing the type of a threat permits estimation of severity of the attack and helps adopt a good step. Existing strategies usually notice malicious URLs of one attack kind. During this paper, we have a tendency to propose methodology using machine learning to notice malicious URLs of all the popular attack varieties and establish the character of attack a malicious address tries to launch. Our method uses a range of discriminative options together with matter properties, link structures, webpage contents, DNS information, and network traffic. Several of those options are novel and extremely effective.

**Keywords :** Cyber Attacks, DNS Information, URL, Malicious Address.

## I. INTRODUCTION

As any file on a computer is to be found by giving its filename, similarly to trace any Web site its Uniform Resource Locators (URLs) are used. One can retrieve a site by typing a URL into the address bar of browser or simply by clicking correct URL.

One can access desired website. E.g. https://mail.google.com/mail/#inboxIt follows standard syntax :< protocol>< hostname>. Malicious Web sites covers a range of different illicit enterprises which are unsafe to visit, that's why different types of malicious sites allocate various threats to users. If type of this threat is known it will be easy to inspect these types independently and understand their features which will be helpful to track the malicious site and to find out solution against a particular kind of threat. Three major categories of malicious sites (Spamming, Phishing, and Malware) are considered in this paper, and each class is separated from the other by level of interaction required by the user. A simple probabilistic classifier based on applying Bayes theorem from Bayesian statistics with strong naïve independence assumptions is known as Naive Bayes classifier. In more detail the fundamental probability.

Model is described as "independent feature model". In simple terms, a Naive Bayes classifier assumes that the presence or absence of a particular feature of a class is unrelated to the presence or absence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 2.5" in diameter. Each property has its independent contribution to the probability that this fruit is an apple, Even if these features depend on each other or upon the existence of the other features. For supervised learning Process Naive Bayes classifiers can be used for training it works very efficiently, with the help of precise characteristics of the probability model.

Naïve Bayes Classifier technique is mostly preferred when the dimensionality of the inputs is high. In spite of simplicity of Naive Bayes, it can handle and perform better than more complicated classification methods. Naïve Bayes model can be used for identifying the patients having heart disease by determining characteristics of patients. It calculates the probability of each input attribute independently for the expected state. Maximum likelihood method is used by many real time applications for parameter estimation, it can work without making an allowance for or using any Bayesian methods. In 2004, work on analysis of the Bayesian classification problem demonstrates that it shows outstanding performance by giving some theoretical causes for effectiveness of Naive Bayes classifiers. After two year i.e. in 2006 it is analyzed that Bayes classification is outperformed by supplementary approaches for e.g. boosted trees or random forests. After broad comparison it is concluded that small amount of data is enough for training purpose. For classification purpose it is mandatory to calculate means and variances of the variables. As independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix. Spamming, phishing, and malware. Beginning with an overview of the classification problem, for which trained datasets are used as a collection of URLs, followed by a discussion of the learning approaches used for classification on basis of features, and finally SVM and Naïve Bayes are classifier used for the URL classification.

## II. PROPOSED SYSTEM

Our method consists of three stages as shown in Figure 1: training data collection, supervised learning with the training data, and malicious URL detection and attack type identification. These stages can operate sequentially as in batched learning, or in an interleaving manner: additional data is collected to incrementally train the classification models while the models are used in detection and identification. Interleaving operations enable our method to adapt and improve continuously with new data, especially with online learning where the output of our method is subsequently labeled and used to train the classification models.

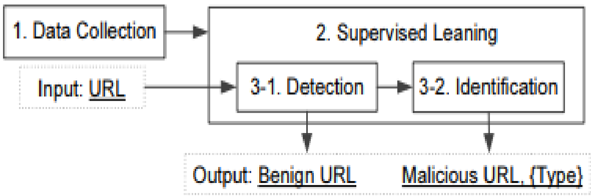


**Figure 1.** The framework of our method.

## III. LEARNING ALGORITHMS

The two tasks performed by our method, detecting malicious URLs and identifying attack types, need different machine learning methods. The first task is a binary classification problem. The Support Vector Machine (SVM) is used to detect malicious URLs. The second task is a multi-label classification problem. Two multi-label classification methods, (RAkELand ML-kNN), are used to identify attack types.

**Task1: Support Vector Machine (SVM).** SVM is a widely used machine learning method introduced by Vapnik et al.. SVM constructs hyperplanes in a high or infinite dimensional space for classification. Based on the Structural Risk Maximization theory, SVM finds the hyperplane that has the largest distance to the nearest training data points of any class, called functional margin. Functional margin optimization can be achieved by maximizing the following equation

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

Subject to

$$\sum_{i=1}^{n} \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, ..., n$$

Whereαi and αj are coefficients assigned to training samples xi and xj .K (xi, xj) is a kernel function used to measure similarity between the two samples. After specifying the kernel function, SVM computes the coefficients which maximize the margin of correct classification on the training set. C is a regulation parameter used for tradeoff between training error and margin, and training accuracy and model complexity.

**Task2: RAkEL. And ML-kNN**. RAkEL is a high-performance multi-label learning method that accepts any multi-label learner as a parameter. RAkEL creates m random sets of k label combinations, and builds an ensemble of Label Powerset (LP) classifiers from each of the random sets. LP is a transformation-based algorithm that accepts a single-label classifier as a parameter. It considers each distinct combination of labels that exists in the training set as a different class value of a single-label classification task. Ranking of the labels is produced by averaging the zero-one predictions of each model per considered label. An ensemble voting process under a threshold t is then employed to make a decision for the final classification set. We use C4.5 as the single-label classifier and LP as a parameter of the multi-label learner.

ML-kNN is derived from the traditional k-Nearest Neighbor (kNN) algorithm. For each unseen instance, its k nearest neighbors in the training set is first identified. Based on the statistical information gained from the label sets of these neighboring instances, maximum a posteriori principle is then utilized to determine the label set for the unseen instance.

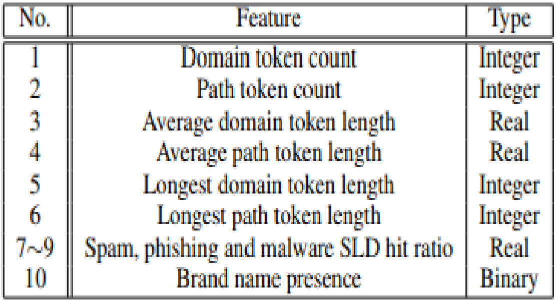
## IV. Discriminative Features

Our method uses the same set of discriminative features for both tasks: malicious URL detection and attack type identification. These features can be classified into six groups: lexicon, link popularity, webpage content, DNS, DNS fluxiness, and network traffic. They can effectively represent the entire multifaceted properties of a malicious URL and are robust to known evasion techniques.

### A. Lexical Features

Malicious URLs, esp. those for phishing attacks, often have distinguishable patterns in their URL text. Ten lexical features, listed in Table 1, are used in our method. Among these lexical features, the average domain/path token length (delimited by ‘.’, ‘/’, ‘?’, ‘=’, ‘-’, ‘ ’) and brand name presence were motivated from a study by McGrath and Gupta that phishing URLs show different lexical patterns. For example, a phishing URL likely targets a widely trusted brand name for spoofing, thus contains the brand name. Therefore, we employ a binary feature to check whether a brand name is contained in the URL tokens but not in its SLD (Second Level Domain) 1.

Table 1: Lexical features (LEX)



| No. | Feature | Type |
|---|---|---|
| 1 | Domain token count | Integer |
| 2 | Path token count | Integer |
| 3 | Average domain token length | Real |
| 4 | Average path token length | Real |
| 5 | Longest domain token length | Integer |
| 6 | Longest path token length | Integer |
| 7~9 | Spam, phishing and malware SLD hit ratio | Real |
| 10 | Brand name presence | Binary |

In our method, the detection model maintains two lists of URLs: a list of benign URLs and a list of malicious URLs. The identification model breaks the list of malicious URLs into three lists: spam, phishing, and malware URL lists. For a URL, our method extracts its SLD and calculates the ratio of the number that the SLD matches SLDs in the list of malicious URLs or a list of specific type of malicious URLs (e.g., spam URL list) to the number that the SLD matches SLDs in the list of benign URLs. This ratio is called the malicious or a specific attack type (e.g., spam) SLD hit ratio feature, which is actually an a priori probability of the URL to be malicious or of a specific malicious type (e.g., spam) based on the precompiled URL lists.

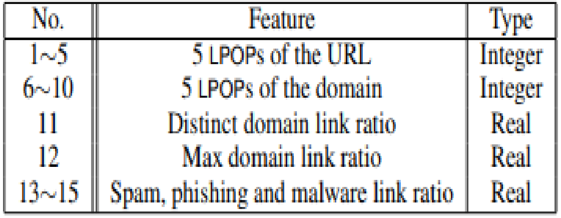
Previous methods use URL tokens as the “bag-of words” model in which the information of a token’s position in a URL is lost. By examining a large set of malicious and benign URLs, we observed that the position of a URL token also plays an important role. SLDs are relatively hard to forge or manipulate than URL tokens at other positions. Therefore, we discard the widely used “bag-of-words” approach and adopt several new features differentiating SLDs from other positions, resulting in a higher robustness against lexical manipulations by attackers. Lexical features No. 1 to No. 4 in Table 1 are from previous work. Feature No. 10 is different from the “bag-of-words” model used in previous work by excluding the SLD position. The other lexical features in Table 1 are novel features never used previously.

### B. Link Popularity Features

One of the most important features used in our method is “link popularity”, which is estimated by counting the number of incoming links from other WebPages. Link popularity can be considered as a reputation measure of a URL. Malicious sites tend to have a small value of link popularity, whereas many benign sites, especially popular ones, tend to have a large value of link popularity. Both link popularity of a URL and link popularity of the URL's domain are used in our method. Link popularity (LPOP) can be obtained from a search engine2. Different search engines may produce different link popularity due to different coverage of WebPages each has crawled. In our method, five popular search engines, AltaVista, AllTheWeb, Google, Yahoo!, and Ask, are used to calculate the link popularity of a URL and the link popularity of its domain, corresponding to LPOP features No. 1 to 10 in Table 2.

One problem in using link popularity is "link farming", a link manipulation that uses a group of WebPages to link together. To address this problem, we develop five additional LPOP features by exploiting different link properties between link-manipulated malicious websites and popular benign websites. The first feature, the distinct domain link ratio, is the ratio of the number of unique domains to the total number of domains that link to the targeted URL. The second feature, the max domain link ratio, is the ratio of the maximum number of links from a single domain to the total number of domains that link to the targeted URL. Link manipulated malicious URLs tend to be linked many times with a few domains, resulting in a low score on the distinct domain link ratio and a high score on the max domain link ratio. A study by Castillo et al. indicates that spam pages tend to be linked mainly by spam pages. We believe that a hypothesis to assume that not only spam pages, but also phishing and malware pages tend to be linked by phishing and malware pages, respectively, is plausible. Therefore, we develop the last three features: spam link ratio, phishing link ratio, and malware link ratio. Each represents the ratio from domains of a specific malicious type that link to the targeted URL. To measure these three features, we use the malicious URL lists described in Section 3.1. The link popularity features described in this subsection are all novel features.

Table 2: Link popularity features (LPOP)

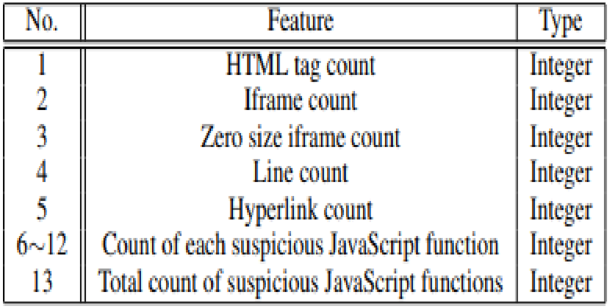


| No. | Feature | Type |
|---|---|---|
| 1~5 | 5 LPOPs of the URL | Integer |
| 6~10 | 5 LPOPs of the domain | Integer |
| 11 | Distinct domain link ratio | Real |
| 12 | Max domain link ratio | Real |
| 13~15 | Spam, phishing and malware link ratio | Real |

## C. Webpage Content Features

Recent development of the dynamic webpage technology has been exploited by hackers to inject malicious code into WebPages through importing and thus hiding exploits in webpage content. Therefore, statistical properties of client-side code in the Web content can be used as features to detect malicious WebPages. To extract webpage content features (CONTs), we count the numbers of HTML tags, iframes, zero size iframes, lines, and hyperlinks in the webpage content. We also count the number for each of the following seven suspicious native JavaScript functions: escape(), eval(), link(), unescape(), exec(), link(), and search() functions. As suggested by a study of Hou et al., these suspicious JavaScript functions are often used by attacks such as cross-site scripting and Web-based malware distribution. For example, unescape () can be used to decode an encoded shell code string to obfuscate exploits. The counts of these seven suspicious JavaScript functions form features No. 6 to No. 12 in Table 3. The last feature in this table is the sum of these function counts, i.e., the total count of these suspicious JavaScript functions. All the features in Table 3 are from the previous work.

Table 3: Webpage content features (CONT)



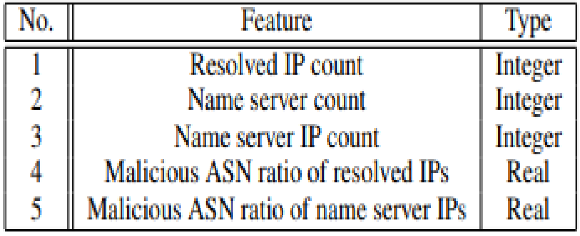
| No. | Feature | Type |
|---|---|---|
| 1 | HTML tag count | Integer |
| 2 | Iframe count | Integer |
| 3 | Zero size iframe count | Integer |
| 4 | Line count | Integer |
| 5 | Hyperlink count | Integer |
| 6~12 | Count of each suspicious JavaScript function | Integer |
| 13 | Total count of suspicious JavaScript functions | Integer |

The CONTs may not be effective to distinguish phishing websites from benign websites because a phishing website should have similar content as the authentic website it targets. However, this very nature of being sensitive to one malicious type but insensitive to other malicious types is very much desired in identifying the type of attack that a malicious URL attempts to launch.

## D. DNS Features

The DNS features are related to the domain name of a URL. Malicious websites tend to be hosted by less reputable service providers. Therefore, the DNS information can be used to detect malicious websites. Ramachandran et al. showed that a significant portion of spammers came from a relatively small collection of autonomous systems. Other types of malicious URLs are also likely to be hosted by disreputable providers. Therefore, the Autonomous System Number (ASN) of a domain can be used as a DNS feature.

Table 4: DNS features (DNS)



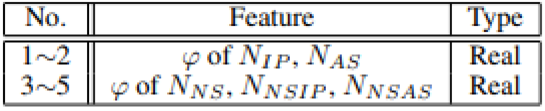
| No. | Feature | Type |
|---|---|---|
| 1 | Resolved IP count | Integer |
| 2 | Name server count | Integer |
| 3 | Name server IP count | Integer |
| 4 | Malicious ASN ratio of resolved IPs | Real |
| 5 | Malicious ASN ratio of name server IPs | Real |

All the five DNS features listed in Table 4 are novel features. The first is the number of IPs resolved for a URL's domain. The second is the number of name servers that serves the domain. The third is the number of IPs these name servers are associated with. The next two features are related to ASN. As we have mentioned in Section 3.1, our method maintains a benign URL list and a malicious URL list. For each URL in the two lists, we record its ASNs of resolved IPs and ASNs of the name servers. For a URL, our method calculates hit counts for ASNs of its resolved IPs that matches the ASNs in the malicious URL list. In a similar manner, it also calculates the ASN hit counts using the benign URL list. Summation of malicious ASN hit counts and summation of benign ASN hit counts are used to estimate the malicious ASN ratio of resolved IPs, which is used as an a priori probability for the URL to be hosted by a disreputable service provider based on the precompiled URL lists. ASNs can be extracted from MaxMind's database file.

### E. DNS Fluxiness Features

A newly emerging fast-flux service network (FFSN) establishes a proxy network to host illegal online services with a very high availability. FFSNs are increasingly employed by attackers to provide malicious content such as malware, phishing websites, and spam campaigns. To detect URLs which are served by FFSNs, we use the discriminative features proposed by Holz et al., as listed in Table 5.

Table 5: DNS fluxiness features (DNSF)



| No. | Feature | Type |
|---|---|---|
| 1~2 | $\varphi$ of $N_{IP}$, $N_{AS}$ | Real |
| 3~5 | $\varphi$ of $N_{NS}$, $N_{NSIP}$, $N_{NSAS}$ | Real |

We look up the domain name of a URL and repeat the DNS lookup after TTL (Time-To-Live value in a DNS packet) timeout given in the first answer to have consecutive lookups of the same domain. Let NIP and NAS be the total number of unique IPs and ASNs of each IP, respectively, and NNS, NNSIP, NNSAS be the total number of unique name servers, name server IPs, and ASNs of the name server IPs in all DNS lookups. Then, we can estimate fluxiness using the acquired numbers. For example, fluxiness of the resolved IP address is estimated as follows

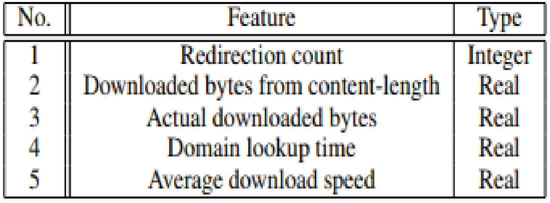
$$\varphi = N_{IP}/N_{single},$$

Where φ is the fluxiness of the domain and Nsingle is the number of IPs that a single lookup returns. Similarly, all of the other fluxiness features are estimated.

### F. Network Features

Attackers may try to hide their websites using multiple redirections such as iframe redirection and URL shortening. Even though also used by benign websites, the distribution of redirection counts of malicious websites is different from that of redirection counts of benign websites. Therefore, redirection count can be a useful feature to detect malicious URLs. In a HTTP packet, there is a content-length field which is the total length of the entire HTTP packet. Hackers often set malformed (negative) content-length in their websites in a buffer overflow exploit. Therefore, content-length is used as a network discriminative feature. Benign sites tend to be more popular with a better service quality than malicious ones. Web technologies tend to make popular websites quick to look up and faster to download. In particular, benign domains tend to have a higher probability to be cached in a local DNS server than malicious domains, esp. those employing FFSNs and dynamic DNS. Therefore, domain lookup time and average download speed are also used as features to detect malicious URLs. The network features listed in Table 6 except the third and fifth features are novel features.

Table 6: Network features (NET)



| No. | Feature | Type |
|---|---|---|
| 1 | Redirection count | Integer |
| 2 | Downloaded bytes from content-length | Real |
| 3 | Actual downloaded bytes | Real |
| 4 | Domain lookup time | Real |
| 5 | Average download speed | Real |

## V. CONCLUSION

The Web has become associate economical channel to deliver numerous attacks like spamming, phishing, and malware. To thwart these attacks, we've given a machine learning methodology to each observe malicious URLs and establish attack sorts. We've given numerous sorts of discriminative options no heritable from lexical, webpage, DNS, DNS fluxiness, network, and link quality properties of the associated URLs. Several of those discriminative features like link quality, malicious SLhit ratio, malicious link ratios, and malicious ASN ratios are novel and extremely effective, as our experiments found out. SVM was accustomed observe malicious URLs, and both RAkEL and ML-kNN were accustomed establish attack types. Our experimental results on real-life information showed that our methodology is extremely effective for each detection and identification tasks. Our methodology achieved associate accuracy of over ninety eight in detective work malicious URLs associated an accuracy of over ninety three in characteristic attack sorts. Additionally, we studied the effectiveness of every cluster of discriminative features on each detection and identification, and discussed evadability of the options.

## VI. REFERENCES

[1]. Xiangnan Kong, Michael K. Ng, and Zhi-Hua Zhou, “Transductive Multi-label Learning via Label Set Propagation” IEEE Transactions On Knowledge And Data Engineering, Vol. 25, No. 3, March 2013.

[2]. GrigoriosTsoumakas, IoannisKatakis,“Multi-Label Classification: An Overview” International Journal Data Warehousing and Mining , 2007.

[3]. Lei Wu, Min-Ling Zhang “Multi-Label Classification with Unlabeled Data: An Inductive Approach” JMLR: Workshop and Conference proceedings 29:197-212, 2013

[4]. Charles X. Ling, Victor S. Sheng “Cost-Sensitive Learning and the Class Imbalance Problem” Encyclopedia of Machine Learning. C. Sammut (Ed.). Springer.

[5]. Hung-Yi Lo, Shou-De Lin, and Hsin-Min Wang, “Generalized k-Label sets Ensemble for Multi-Label and Cost-Sensitive Classification” IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 7, July 2014 1679 Chun-Liang Li, Hsuan-Tien Lin “Condensed Filter for Cost-sensitive Multi-label Classification” International conference on machine learning , China JMLR :W/P volume 32.

[6]. M.-L. Zhang and Z.-H. Zhou, “ML-kNN: A Lazy Learning Approach to Multi-Label Learning,” Pattern Recognition, vol. 40,no. 7, pp. 2038-2048, 2007. Pranali Dhongade et al, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.12, December-2014, pg. 189-196 © 2014, IJCSMC All Rights Reserved 195

[7]. R.E. Schapire and Y. Singer, “BoostTexter: A Boosting-Based System for Text Categorization,” Machine Learning, vol. 39, nos. 2/3, pp. 135-168, 2000

[8]. Y. Freund and R.E. Schapire, “A Decision-Theoretic Generalization of On- Line Learning and an Application to Boosting,” J. Computer and System Sciences, vol. 55, no.1, pp. 119-139, 1997.

[9]. N. Ghamrawiand A. McCallum, “Collective Multi-Label, Classification”Proc. 14th Int’l

Conf. Information and Knowledge Management, pp. 195-200, 2005

[10]. Elisseeff and J. Weston, “A Kernel Method for Multi-Labelled Classification,” Advances in Neural Information Processing Systems 14, T.G. Dietterich, S. Becker and Z. Ghahramani, eds., pp. 681-687, MIT Press, 2002.

[11]. V.N. Vapnik, Statistical Learning Theory. Wiley, 1998.

[12]. T. Joachims, “Transductive Inference for Text Classification Using Support Vector Machines,” Proc. 16th International Conf. Machine Learning, pp. 200-209, 1999.

[13]. Krzysztof Dembczy´, Weiwei Cheng ,Eyke H¨ullermeier1 “ Bayes Optimal Multi-labelClassification via Probabilistic Classifier Chains “International Conference on Machine Learning, Haifa, Israel, 2010.

[14]. HiteshriModi Mahesh Panchal, “Experimental Comparison of Different Problem Transformation Methods for Multi-Label Classification using MEKA” International Journal of Computer Applications Volume 59No.15, December 2012

[15]. Oscar Luaces, Jorge Díez, José Barranquero· Juan José del Coz · Antonio Baham “Binary relevance efficacy for multilabel classification” © Springer-Verlag Berlin Heidelberg 2012

[16]. Erica Akemi Tanaka1 and Jos´e Augusto Baranauskas “An Adaptation of Binary Relevance for Multi-Label Classification applied to Functional Genomics” ISSN -2012.

[17]. Cherman, E. A., J. Metz and M. C. Monard, “Incorporating label dependency into the binary relevanceframework for multi-label classification, Expert Systems with Applications” 39(2012), pp. 1647–1655.

[18]. Newton Spolaor,EvertonAlvaresCherman, Maria Carolina Monard&Huei Diana Lee, “A Comparison of Multilabel Feature Selection Methods using the Problem Transformation Approach” ELSEVIER-Electronic Notes in Theoretical Computer Science 292 (2013) 135–151

[19]. GrigoriosTsoumakas, IoannisKatakis, and IoannisVlahavas, “Mining Multi-label Data” Data Mining and Knowledge Discovery Handbook 2010, pp 667-685

[20]. D. W. Aha, Lazy learning: Special issue editorial, Artificial Intelligence Review 11 (1-5) (1997) 7-10.

## AUTHOR’S PROFILE

R. Hamsaveni working as an Assistant Professor in Sri Venkateswara College of engineering and technology, Chittoor, A.P.

A.Hariprasad Reddy received the P.G degree from Sri Venkateswara College of engineering and technology, Chittoor, A.P in 2018.

C.Kesavalu received the P.G degree from Sri Venkateswara College of engineering and technology, Chittoor, A.P in 2018.