Performance Analysis of Classification Techniques for Suspicious URL Detection in Social Networks

Maharshi Tiwari
Samrat Ashok Technological Institute
Department of Information Technology,
Vidisha, M.P., India

Abhishek Mathur
Samrat Ashok Technological Institute
Department of Information Technology,
Vidisha, M.P., India

ABSTRACT
Social network services (SNSs) are increasing popular. Now a day's most of the people in all over the world use Facebook, twitter for sharing their ideas. Though suspicious users collectively use by them to embed to harmful activities that may be tricky in securing user's personal information and data. This is challenge for social networks to rectify this type of security breach. The social networks or community websites must be able to identify phishing and suspicious urls. Machine learning techniques are proved an efficient tool in classifying benign and the suspicious urls from the set of many urls most of the solutions for training the classification models that supported all totally different sorts of feature sets. However, the most of the solutions does not provide good results as we expect from them on the basis of performance, behavior and some other criteria.
In this study, a feature set is presented that combines the features of traditional heuristics and social networking. Furthermore, a suspicious URL identification system for use in social network environments is proposed which is based on comparative study of three algorithms named as Bayesian classification, KNN, SVM. The experimental results indicate that the proposed approach achieves a high detection rate.

Keywords
ON-Line Social Networks, Suspicious URLS, Naive Bayesian Classification, KNN, SVM, Machine Learning,

1. INTRODUCTION
Social Network is the investigation of social substances (individuals in an organization, called performing artists), and their associations and connections. The collaborations and connections can be spoken to with a system or diagram, where each vertex (or hub) speaks to a performing artist and every connection speaks to a relationship. From the system we can ponder the properties of its structure, and the role, position and glory of every social performer. We can likewise discover different kinds of sub-charts, e.g., groups shaped by gatherings of performing artists.

Social Network investigation is valuable for the Web on the grounds that the Web is basically a virtual society, and along these lines a virtual social network, where every page can be viewed as a social on-screen character and every hyperlink as a relationship. Large portions of the outcomes from social networks can be adjusted and stretched out for utilization in the Web connection. The thoughts from social network investigation are undoubtedly instrumental to the execution of web inquiry apparatuses [1].

Person to person communication sites are used to convey and for communicating their hobbies with others in on the web. It gives simple entry to new patterns/points and quicker correspondence over longer separations. Some of Internet clients use SNS for meeting new companions. A few clients use it to discover old companion and relatives. SNS give clients heaps of advantages like sharing different level of data, media sharing (photograph, feature, get) and numerous different things. To most astounding degree SNS additionally permits you to make your gathering taking into account you're intrigue. It is a simple approach to discover companions and is likes "one to numerous "or "numerous to numerous" connections [2].

1.1 Privacy Issues
In this segment, two security issues will be talked about. To begin with his client's obscurity or client's character. Two methodologies of distinguishing client's personalities in online social networks will be depicted. The second issue is client's profile and individual data spillage.

1.1.1 Users' Anonymity
Clients utilize their genuine name to speak to their records in almost every social networking based websites. Along these lines, their personality is presented freely to other interpersonal organization clients, and additionally others in the online world. Additionally, informal organization client's record can be listed via web search tool and typically showed up in the top rank of the indexed lists. For this situation, if assailants know the name of the casualties, they can without much of a stretch quest for casualty's profile, or they can seek through informal communication locales to get new casualties. The two strategies that will be talked about are de-anonymize assault and neighborhood assault.

1.1.2 De-Anonymization Attack:-
Gilbert Wondracek and his group demonstrated that by utilizing gathering participation data and history taking system, aggressors could uncover secrecy of social networks clients [3].

In this system, what assailants need to realize is in which social networks (gathering of clients that has comparative intrigues or gathering of individuals with same foundation e.g. went to same school or work at the same spot) casualties fit in with. The social networks gathering is being engaged following the quantity of a social networks singular client is a ton bigger than the quantity of gatherings in informal organizations. Consequently, it is simpler to first concentrate on the gathering, and afterward utilize the gathering to get to individual client. Aggressors will utilize history-taking strategy to acquire which URLs (sites) that casualties went to in the past to discover casualty's gathering.

There are two sorts of connections in social networking based sites. A static connection is the same for all social networking based sites clients. It is utilized for showing client's home

segment, and a dynamic connection that contains some data extraordinary to every client or every gathering [4].

1.1.3 Neighborhood Attack

Social networking based sites can be represented by amusing blueprint area a bulge represents an amusing arrangement user, and a bend represents accord amid two amusing arrangement users. Neighborhood advance is based on the abstraction that if attackers apperceive the neighbors of the victims' node, and the accord amid them, again attackers can analyze victims' node [5].

1.2 Examples of Suspicious Sites in Twitter

1.2.1 Blackraybansunglasses.com

1. The blackraybansunglasses.Com is a suspicious website associated with unsolicited mail tweets this web site evaluates the sort of person whether its viewers are normal browsers or crawlers. It redirects the common browsers to random junk mail pages and redirects the crawlers to google.Com stopping crawler from reaching to junk mail pages.

2. a different important factor is that this web page uses typical Twitter API which isn't utilized by developed spammers due to the fact that if they use this API. Spam detection system can differentiate the suspicious tweet and ordinary tweets.

1.2.2 24newspress.com

This website online does no longer perform conditional redirection to hinder investigation. As an alternative, it makes use of a quantity of IP addresses, domain names quantity of one of a kind shortened URLs and different Twitter accounts to distribute tweets to twitter customers. Furthermore, it misapplies the cell Twitter internet interface to deliver its unsolicited mail tweets [6].

2. LITERATURE SURVEY

In this paper [7] author has proposed a method for twitter stream where, Twitter can experience from the effects of malicious tweets that contain suspicious URLs like malware, phishing and spam circulation.. Here they propose a suspicious URL identification system for Twitter, WARNINGBIRD. Not at all like the past systems, is WARNINGBIRD powerful when ensuring against restrictive redirection, since it doesn't depend on the features of malicious landing pages that might not be accessible. They presented new features on the premise of these relationships, implemented a real-time classification system utilizing these features, and calculate the system's performance and accuracy. The experimental results demonstrated that their system is profoundly accurate and deployed as a real-time system for classifying vast tweets samples from the public timeline of twitter.

In this paper [8] author has proposed a method for malicious urls where Malicious URLs have been generally used to mount different cyber crimes that contain spamming, phishing and malware. They propose system utilizing machine learning for figuring out how to distinguish malicious URLs of all the prevalent attack types and recognize the way of attack a malicious URL endeavors to dispatch. Their strategy utilizes a mixed bag of discriminative features that contain textual properties, connection structures, website page substance, DNS information, and network traffic. Their system accomplished an accuracy of more than 98% in distinguishing malicious URLs and an accuracy of more than 93% in recognizing attacks sorts.

In this paper [9] author has proposed that Online social networking based sites (OSNs) are amazingly prominent among Internet clients. They propose to recreate spam posts into campaigns for classification as opposed to analyze them exclusively. They evaluate the system utilizing 187 million wall messages composed from Facebook and **17** million tweets composed from Twitter. In distinctive parameter settings, the true positive rate comes to 80.9% whereas the false positive rate comes to 0.19% in the greatest part. Moreover, it stays accurate for over 9 months after the introductory training stage.

In this paper [10] author has proposed a method for phishing urls Phishing has been simple and efficient way for fraud and trickiness on the web during the solutions, for example, URL blacklisting have been powerful to some degree, their dependence on exact match with the blacklisted sections makes it simple for attackers to evade. They begin with the perception that attackers regularly utilize basic modifications (for example altering higher level domains) to URLs. Their system, Phish Net, exploits this perception utilizing two segments. In the first part, they propose five heuristics to count basic combinations of known phishing websites to find latest phishing URLs. The other segment comprises of an approximate matching algorithm that analyzes a URL into various segments that are coordinated independently against entries in the blacklist. In Their assessment with ongoing blacklist feeds, they found probably eighteen thousand novel phishing URLs from a set of six thousand new blacklist values. They likewise demonstrate that their approximate matching algorithm prompts not very many FPR (false positive rate) (3%) and negatives (5%).

In this paper [11] author has proposed a method for twitter spam detection where, Twitter spam recognition is a latest area of study in which mainly past works had concentrated on the identification of malicious client records and honey pot-based methodologies. Be that as it may, here they display a procedure in view of two new angles: the discovery of spam tweets in isolation and without past information of the client; and the purpose of a statistical analysis of language to recognize spam in drifting themes. Their method achieves estimations of 89.3% and 93.7% in non-spam and spam accurately classifier, and just 6.3% of the non-spam tweets were misclassified as spam. They have also performed a second assessment test with another set of unlabeled tweets and a group of assessors so as to a further assessment of the method. Assessors reasoned that the spam identification method had the capacity distinguish a 94.5% of spam tweets and get a false positive rate of 5.4%.

3. PROPOSED ALGORITHM

3.1 Naïve Bayesian Classification

The working of the naïve Bayesian classifier, or basic Bayesian classifier, is as follows:

1. Suppose a set of training of tuples is D and their related class labels. Obviously, every tuple is represented through an n-dimensional attribute vector, $X = (x_1, x_2, \ldots\ldots x_n)$, delineating n estimations prepared on the tuple from n attributes, correspondingly, $A_1, A_{2.} \ldots\ldots A_n$.

2. Assume that there are m classes, $C_1, C_2 \ldots\ldots C_m$. Given a tuple, X, the classifier will predict that X fits in with the class having the most elevated posterior probability, moulded on X. That is, the naive Bayesian classifier

predicts that tuple X has a place with the class C_i if and only if

$$P(C_i \mid X) > P(C_j \mid X) \quad for I \le J \le m, j \ne i,$$

Accordingly, we augment $P(C_i \mid X)$. The class (C_i) for which $P(C_i \mid X)$ is expanded is known as the greatest posteriori hypothesis. By Bayes' theorem (

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)} \qquad (1)$$

1. As P(X) is steady for all classes, only $P(X \mid C_i)P(C_i)$ should be augmented. In the event that the class prior probabilities are not identified, then it is usually acknowledged that the classes are presently as possible, that is, $P(C_1) = P(C_2) = ...P(C_m)$ and we would along these lines exploit $P(X \mid C_i)$. Else, we exploit. $P(X \mid C_i)P(C_i)$.Note down that the class prior probabilities may be accessed through $P(C_i) = C_i, D / \mid D \mid$, where $\mid C_i, _D \mid$ is the value of class training tuples C_i in D.

2. Given data sets with numerous attributes, it would be to a great degree computationally costly to register $P(X \mid C_i)$. To diminish calculation in assessing, the naive assumption of class conditional independence is prepared. This presumes that the attributes' qualities are conditionally independent of each other, certain the class label of the tuple (i.e., that there are no reliance connections among the attributes). Along these lines,

$$P(X \mid C_i) = \prod_{k=1}^{a} P(X_k \mid C_i)$$

$$= P(X_1 \mid C_i) \times P(X_2 \mid C_i) \times ..... \times P(X_n \mid C_i) \qquad (2)$$

We can without much of a stretch calculate the probabilities $P(X_1 \mid C_i), P(X_2 \mid C_i), ... P(X_n \mid C_i)$ from the training tuples. Review that here X_k alludes to the estimation of attribute A_k for tuple X. For every attribute, we take a gander at whether the attribute is unconditional or uninterrupted-valued. Case in point, to calculate $P(X \mid C_i)$ we focus on these points:

1. If A_k is unconditional, then $P(X_k \mid C_i)$ is the quantity of class tuples is C_i in D having the significance X_k for A_k, alienated $\mid C_i, _D \mid$ by , the quantity of class tuples is C_i in D.

2. In the event that A_k is continuous significance, then we have to do more work, however the figuring is really direct. A continuous significance attribute is normally accepted to have a Gaussian dissemination through a mean μ and standard deviation σ, characterized by

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

So that

$$P(X_k \mid C_i) = g(X_k, \mu C_i, \sigma C_i) \qquad (3)$$

These mathematical equations may seem overwhelming, however hang on! We have to process μC_i and σC_i , that are the mean (i.e., average) and standard deviation, individually, of the estimations of attribute A_k for training tuples of class C_i . We then connect these two amounts to Eq. (8.13), together with X_k , to calculate $P(X_k \mid C_i)$.

For instance, let X= (35, $40,000), wherever A_1 and A_2 are the attributes period and salary, individually. Assume the class label attribute be buys_ personal computer. The related class label for X is yes (i.e., buys_ personal computer = yes). How about we assume that period has not been discredited and along these lines exists as a continuous-valued attribute. Assume that from the training set, we find that clients in D who purchase a PC 38+12 are years old. Similar, for attribute period and this class, we have $\mu = 38$ years and $\sigma = 12$. We can block these amounts, alongside $X_1 = 35$ for our tuple X, into Eq. (8.13) to calculate P (age = 35 | purchases _ personal computer =yes). For a snappy audit of mean and standard deviation estimates, that is shown in Section 2

5 For predicting X the class label $P(X \mid C_i)P(C_i)$ is assessed for every class C_i . The given classifier predicts the class label of tuple X is the class C_i if and only if

$P(X \mid C_i)P(C_i)) > P(X \mid C_j)P(C_j)\ for i \le j \le m, j \ne i.$ In other way, the predicted class label is the class C_i for which P $P(X \mid C_i)P(C_i)$ is the most extreme. Different exact investigations of this classifier in examination to decision tree and neural network classifiers have observed it to be comparable in a few areas. In principle, Bayesian classifiers have the less error rate in correlation to every single other classifier. Nonetheless, by and by this is not generally the situation, inferable from mistakes in the suspicions made for its utilization, for example, class-conditional independence, and the absence of accessible probability data.

Bayesian classifiers are likewise helpful in that they give a hypothetical support to different classifiers that don't expressly utilize Bayes' theorem. For instance, under specific suspicions, it can be demonstrated that numerous neural network and curve-fitting algorithms yield the most extreme posteriori hypothesis; the naive Bayesian classifier does the same.

3.2 Support Vector Machine (SVM)

SVM is a set of supervised learning technique with associated with learning algorithms that is utilized for classification, clustering and regression. SVM also called support vector networks. SVM is a supervised learning algorithm that is utilized for analysis of data and reorganization of patterns, used for analysis of classification and regression of data. It has been shown by several researchers that SVM is also accurate algorithm for classification. It is also widely utilize in Websites page classification and bio-informatics applications.

It has been shown by several researchers that SVM is also accurate algorithm for classification. It is also widely utilize in Websites page classification and bio-informatics applications.

SVM has been functional with achievement to information retrieval problem. SVM a kind of huge boundary Classifier. SVM is machine learning technique which is based on vector space where the purpose is to establish a decision edge between two classes which is maximally for a form a few position in the training data.

$$D = \{(X_i, y_i) | X_i \in R^P, y_i \in \{-1, 1\}\}_{i=1}^{n}$$

Where the value of yi is belongs between 1 and −1, representing the class to which the point xi belongs. Here every xi is a p-dimensional actual vector and we discover the highest-margin hyper plane that divides the points having yi =1 from those having yi = -1.

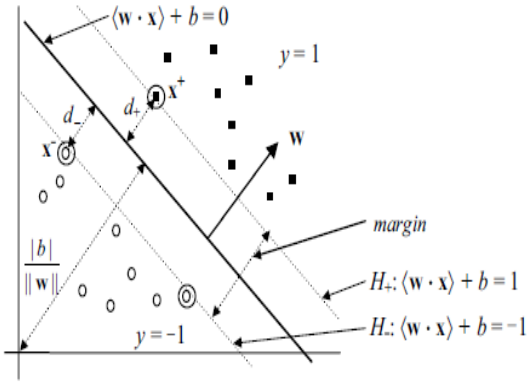

Figure 4.2: Basic Architecture of SVM

Figure 4.2 shows the basic architecture of SVM. Highest margin hyper plane and margins for an SVM with samples from two classes. Samples on the margin are known as support vectors.

Any hyper plane can be described as the set of points x fulfilling. A separating hyper plane is described by the regular vector w and the offset b:

$$w.x + b = 0$$

Where • denotes the dot product. W is also known as the regular vector of the hyper plane. Exclusive of change the regular vector w, unstable b moves the hyper plane parallel to itself. While SVM maximizes the margin between positive and negative data points, let us discover the margin. Let d_+ (correspondingly d_-) be regular the shortest distance from the extrication hyper plane (<w. x>+ b = 0) to the closest positive (negative) data position. The margin of the extrication hyper plane is $d_+ + d_-$.

Let us consider a positive data point (x+, 1) and a negative (x-, -1) which is very close to the hyper plane <w. x> + b = 0. We describe two parallel hyper planes (H+ and H-) that pass by x+ and x- correspondingly. H+ and H- are also parallel to <w. x> + b = 0. We can rescale w and b to achieve

$$H_+ : < w.x^+ > + b = 1$$

$$H_- : < w.x^- > + b = -1$$

The space between the two margin hyper planes H_+ and H_- is ($d_+ + d_-$). Distance from a point x_i to a hyper plane <w. x> + b = 0 is:

$$\frac{|<w.x_i>+b|}{||w||} \quad (4)$$

Therefore, the decision edge <w. x> + b = 0 lies half way between H_+ and H_-. The margin is Therefore

$$margin = d_+ + d_- = \frac{2}{||w||} \quad (5)$$

Consider training sample$\{(x_i, d_i)\}$, where x_i is the input sample, d_i is the preferred output

$$W_0^T X_i + b_0 \geq +1, for\ d_i = +1 \quad (6)$$

$$W_0^T X_i + b_0 \leq -1, for\ d_i = -1 \quad (7)$$

3.3 K-Nearest-Neighbor Classifiers

The k-nearest-neighbor technique was initially portrayed in the mid 1950s. The technique is work serious when given extensive training sets, and did not pick up popularity until the 1960s when expanded calculating power got to be accessible. It has subsequent to been generally utilized as a part of the pattern recognition.

Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it. The training tuples are described by *n* attributes. Each tuple represents a point in an *n*-dimensional space. In this way, all the training tuples are stored in an *n*-dimensional pattern space. When given an unknown tuple, a *k*-nearest-neighbor classifier searches the pattern space for the *k* training tuples that are closest to the unknown tuple. These *k* training tuples are the *k* "nearest neighbors" of the unknown tuple.

"Closeness" is defined in terms of a distance metric, such as Euclidean distance. The Euclidean distance between two points or tuples, say, $X_1 = (x_{11}, x_{12} .... x_{1n})$ and $X_2 = (x_{21}, x_{22} .... x_{2n})$, is

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \quad (8)$$

In other way, for every numeric attribute, we take the distinction between the relating estimations of that attribute in tuple and in tuple, square this distinction, and aggregate it. The square root is taken of the aggregate amassed distance count. Regularly, we standardize the estimations of every quality before utilizing Eq. (9.22). Helps prevent attributes with at first expansive reaches (e.g., salary) from exceeding attributes with at first littler extents (e.g., binary attributes). Min-max standardization, for instance, can be utilized to change a value v of a numeric attribute A to v' in the extent [0, 1] by calculating

$$v' = \frac{v - \min_A}{\max_A - \min_A} \quad (9)$$

Where *minA* and *maxA* are the least amount and greatest values of attribute *A*. Chapter 3 defines different techniques for data standardization as a form of data transformation

For k-nearest-neighbor classification, the obscure tuple is appointed the most commonly identified class between its k nearest neighbors. At the position when k = 1, the obscure tuple is appointed the class of the training tuple that is nearest to it in pattern space. Nearest neighbor classifiers can likewise be utilized for numeric prediction, that is, to give back a actual estimated prediction for a specified obscure tuple. For

this condition, the classifier precedes the normal estimated value of the actual valued labels connected through the k-nearest neighbors of the tuple that is not known.

The past discussion supposes that the attributes utilized to define the tuples which are numeric. For nominal attributes, a straightforward strategy is to analyze the relating estimation of the attribute in tuple with that in tuple. On the off chance that the two are indistinguishable (e.g., tuples X_1 and X_2 both have the shading blue), then the distinction among the two is taken as 0. On the off chance that the two are distinctive (e.g., tuple X_1 is blue yet tuple X_2 is red), then the distinction is thought to be 1. Different techniques may include more modern plans for differential reviewing (e.g., where a bigger distinction score is allocated, state, for blue and white than for blue and black).

By and large, if the estimation of a given attribute A is lost in tuple X_1 and/or in tuple X_2, we expect the most extreme conceivable distinction. Assume that each of the attributes has been mapped to the extent [0, 1]. For ostensible attributes, we take the distinction quality to be 1 if either one or together of the relating estimations are absent. On the off chance that A is numeric and omitted from both tuples X1 and X2, then the distinction is additionally taken to be 1. On the off chance that single estimate is omitted and the another (that we will call) is accessible and consistent, then we can acquire the dissimilarity to be either $|1-v'|$ or $|0-v'|$ (i.e., $1-v'$ or v'), whichever is bigger.

This can be resolved tentatively. Beginning with K=1, we utilize a test set to evaluate the error rate of the classifier. This procedure can be replicated every time by augmenting k to take into account one more neighbors. The k estimate that specifies the minimum error rate might be chose. When all is said in done, the bigger the quantity of training tuples, the bigger the estimation of k will be (so that classification and numeric prediction choices can be founded on a bigger bit of the put away tuples). As the quantity of training tuples approaches infinity and K=1, the error rate can be no more awful than double the Bayes error rate (the recent being the hypothetical least). On the off chance that k additionally approaches infinity; the error rate approaches the Bayes error rate.

Nearest neighbor classifiers utilize distance-based comparisons that characteristically allocate equivalent weight to every attribute. They consequently can experience the ill effects of poor accuracy when given boisterous or immaterial attributes. The strategy, be that as it may, has been altered to join attribute weighting and the pruning of uproarious data tuples. The decision of a distance metric can be basic. The Manhattan (city square) separation (Section 2.4.4), or other distance estimations, might likewise be utilized.

Nearest neighbor classifiers can be to a great degree moderate when classifying test tuples. On the off chance that D is a training database of |D| tuples and K=1, then O (|D|) correlations are required to classify a specified test tuple. By presorting and organizing the put away tuples into search trees, the quantity of correlations can be diminished to O (log |D|). Parallel usage can diminish the running time to a steady, that is, O (t), which is free of |D|.

Different strategies to accelerate classification time incorporate the utilization of distance calculations and altering the put away tuples. In the partial distance strategy, we process the distance based in view of a subset of the n attributes. On the off chance that this distance exceeds a limit, then further calculation for the given put away tuple is ended, and the procedure proceeds onward to the following put away tuple. The altering technique uproots training tuples that demonstrate futile. This technique is additionally alluded to as pruning or gathering on the grounds that it diminishes the aggregate number of tuples put away.

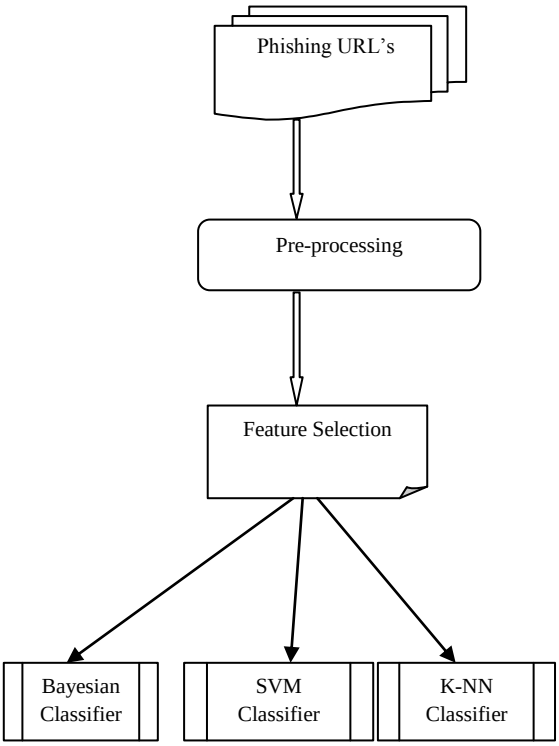

Figure 1: Diagram of Proposed Architecture

4. EXPERIMENTAL RESULTS

In this section, suggest the task of detecting of suspicious urls to calculate the Precision, Recall, Accuracy and F-measure. For evaluation, apply the Bayesian, KNN, and SVM machine learning techniques.

4.1 Datasets Description

The datasets are extracted from UCI machine repository. From there collect the phishing urls data set. This data set contain both type of urls i.e. phishing and benign that contain three classes -1, 1, 0.where -1means phishing urls, 1 means suspicious urls and 0 means benign urls [12].

4.2 Algorithms and Evaluation

To calculate the quality of the algorithms, precision recall and F-measure and accuracy [36] is used, produced by Bayesian, KNN and SVM.

$$F-measure = \frac{2 \times P \times R}{P+R} \tag{10}$$

Where *P and R* are defined as:

$$P(precision) = \frac{TP}{TP+FP} \tag{11}$$

$$R(recall) = \frac{TP}{TP+FN} \tag{12}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (13)$$

For result evaluation perform statistical test and observe that KNN method are considerably superior to type. F-measure Precision and Recall conclude the difference. Table-1, Table-2, Table-3 and Table-4 shows the Results of F-measure, Precision, accuracy and Recall respectively 31 features are taken.

Table-1 Result of Precision for phishing urls dataset

Classification Technique	Train-Test			
	80-20	*70-30*	*60-40*	*50-50*
SVM	0.8104	0.9058	0.5771	0.2790
KNN	0.9929	0.9959	0.8549	0.9981
Bayesian	0.8665	0.8875	0.8594	0.8804

Table-2 Result of Recall for phishing urls dataset

Classification Technique	**Train-Test**			
	80-20	***70-30***	***60-40***	***50-50***
SVM	0.9950	0.9985	1	0.9971
KNN	0.9980	0.9980	0.9984	0.9979
Bayesian	0.9977	0.9977	0.9964	0.9972

Table-3 Result of F-measure for phishing urls dataset

Classification Technique	**Train-Test**			
	80-20	***70-30***	***60-40***	***50-50***
SVM	0.8933	0.9499	0.7319	0.4360
KNN	0.9954	0.9969	0.9928	0.9930
Bayesian	0.9324	0.9394	0.9203	0.9352

Table-3 Result of Accuracy for phishing urls dataset

Classification Technique	**Train-Test**			
	80-20	***70-30***	***60-40***	***50-50***
SVM	0.9141	0.9575	0.8134	0.6820
KNN	0.9959	0.9937	0.9937	0.9938
Bayesian	0.9398	0.9491	0.9346	0.9463

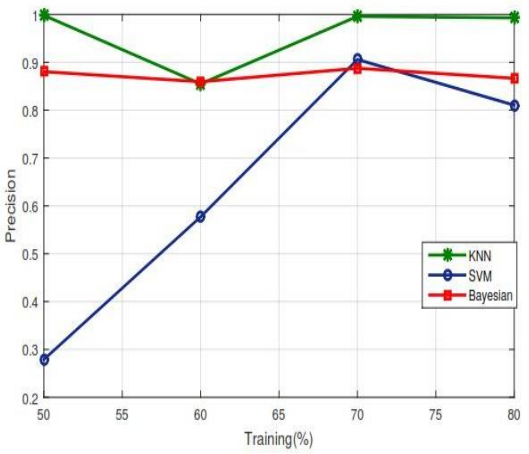

Figure 2: Precision of Bayesian, KNN and SVM for phishing urls dataset

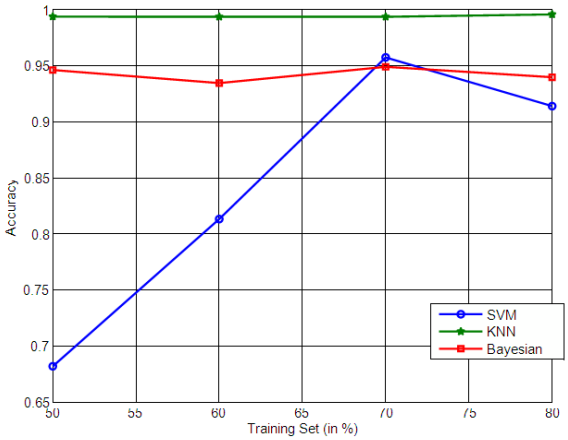

Figure 3: Accuracy of Bayesian, KNN and SVM for phishing urls dataset

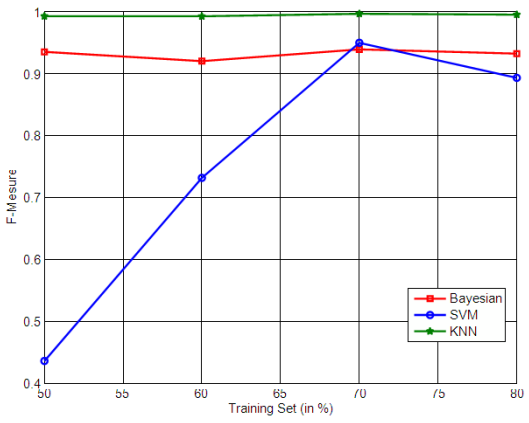

Figure 4: F-measure of Bayesian, KNN and SVM for phishing urls dataset

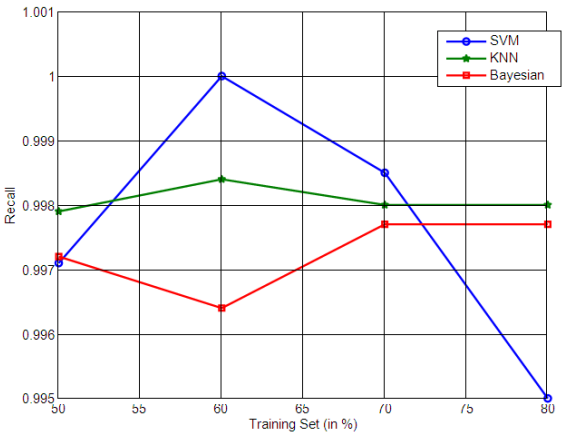

Figure 5: Recall of Bayesian, KNN and SVM for phishing urls dataset

5. CONCLUSION

For Suspicious urls detection the Bayesian classifier is used but experimental evaluation does not provide expected results so we did a comparative study on phishing urls data set where KNN provides the best global results for any data set.

Precision, Recall, Accuracy and F-Measure performance measurements are use to compare the result of Bayesian, KNN and SVM based classifier. The techniques are tested on phishing urls datasets and it found that KNN classifier provides efficient results as compare with Bayesian and SVM classifier. Experimental result shows KNN is best classifier which provides global results for all the data set.

There are further improvements can be done on the performance of this Bayesian, KNN and SVM based classifier. The possible classification techniques that may be used to improve the results of this classifier and boost the work existing in this investigation.

6. REFERENCES

[1] Liu, Bing. *Web data mining: exploring hyperlinks, contents, and usage data*. Springer Science & Business Media, 2007.

[2] Ahmad, Tauseef, and Mohd Mudasir Shafi. "Privacy and security concerns in SNS: a Saudi Arabian users point of view." (2012).

[3] Wondracek, Gilbert, Thorsten Holz, Engin Kirda, and Christopher Kruegel. "A practical attack to de-anonymize social network users." In Security and Privacy (SP), 2010 IEEE Symposium on, pp. 223-238. IEEE, 2010.

[4] Wilson, Jeffrey T., and Mark H. Goldstein. "History-based tracking of user preference settings." U.S. Patent 8,793,614, issued July 29, 2014.

[5] Zhou, Bin, and Jian Pei. "Preserving privacy in social networks against neighborhood attacks." In Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on, pp. 506-515. IEEE, 2008.

[6] Halwar, Jyoti, and Sandip Kadam. "Review on Malicious URL Detection Schemes in Social Networking Site Twitter."

[7] Lee, Sangho, and Jong Kim. "WarningBird: Detecting Suspicious URLs in Twitter Stream." In NDSS. 2012.

[8] Choi, Hyunsang, Bin B. Zhu, and Heejo Lee. "Detecting malicious web links and identifying their attack types." In Proceedings of the 2nd USENIX conference on Web application development, pp. 11-11. USENIX Association, 2011.

[9] Gao, Hongyu, Yan Chen, Kathy Lee, Diana Palsetia, and Alok N. Choudhary. "Towards Online Spam Filtering in Social Networks." In NDSS. 2012.

[10] Prakash, Pawan, Manish Kumar, Ramana Rao Kompella, and Minaxi Gupta. "Phishnet: predictive blacklisting to detect phishing attacks." In INFOCOM, 2010 Proceedings IEEE, pp. 1-5. IEEE, 2010.

[11] Martinez-Romo, Juan, and Lourdes Araujo. "Detecting malicious tweets in trending topics using a statistical analysis of language." Expert Systems with Applications 40, no. 8 (2013): 2992-3000.

[12] Aggarwal, Charu C., and Tarek Abdelzaher. "Integrating sensors and social networks." In Social Network Data Analytics, pp. 379-412. Springer US, 2011.

[13] Zilpelwar, Rashmi A., Rajneeshkaur K. Bedi, and V. M. Wadhai. "An Overview of Privacy and Security in SNS." International Journal of P2P Network Trends and Technology 2, no. 1 (2012).

[14] Marett, Kent, Anna L. McNab, and Ranida B. Harris. "Social networking websites and posting personal information: An evaluation of protection motivation theory." AIS Transactions on Human-Computer Interaction 3, no. 3 (2011): 170-188.

[15] Ellison, Nicole B. "Social network sites: Definition, history, and scholarship."Journal of Computer- Mediated Communication 13, no. 1 (2007): 210-230.

[16] Irani, Danesh, Marco Balduzzi, Davide Balzarotti, Engin Kirda, and Calton Pu. "Reverse social engineering attacks in online social networks." In Detection of intrusions and malware, and vulnerability assessment, pp. 55-74. Springer Berlin Heidelberg, 2011.

[17] Krishnamurthy, Balachander, and Craig E. Wills. "Characterizing privacy in online social networks." In Proceedings of the first workshop on Online social networks, pp. 37-42. ACM, 2008.

[18] Lenhart, Amanda. "Adults and Social Network Websites. Pew Internet and American Life Project." The Pew Center, Washington DC (2009).

[19] Gilbert, Eric, and Karrie Karahalios. "Predicting tie strength with social media." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 211-220. ACM, 2009.

[20] Lampe, Cliff, Nicole B. Ellison, and Charles Steinfield. "Changes in use and perception of Facebook." In Proceedings of the 2008 ACM conference on Computer supported cooperative work, pp. 721-730. ACM, 2008.

[21] Joinson, Adam N. "Looking at, looking up or keeping up with people?: motives and use of facebook." In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp. 1027-1036. ACM, 2008.

[22] Honey, Courtenay, and Susan C. Herring. "Beyond microblogging: Conversation and collaboration via Twitter." In System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on, pp. 1-10. IEEE, 2009.

[23] DiMicco, Joan, David R. Millen, Werner Geyer, Casey Dugan, Beth Brownholtz, and Michael Muller. "Motivations for social networking at work." In Proceedings of the 2008 ACM conference on Computer supported cooperative work, pp. 711-720. ACM, 2008.

[24] Lee, Sangho, and Jong Kim. "WarningBird: Detecting Suspicious URLs in Twitter Stream." In NDSS. 2012.

[25] Chhabra, Sidharth, Anupama Aggarwal, Fabricio Benevenuto, and Ponnurangam Kumaraguru. "Phi. sh/$ oCiaL: the phishing landscape through short URLs." In Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, pp. 92-101. ACM, 2011.

[26] Halwar, Jyoti, and Sandip Kadam. "Review on Malicious URL Detection Schemes in Social Networking Site Twitter.

[27] Dataset URL, "https://archive.ics.uci.edu/ml/datasets/Phishing+Websites".