DOI 10.29042/2019-4791-4796

Helix Vol. 9 (1): 4791- 4796

A Revisit to Classification Algorithms

[1]Chandaka Bhavani Sai Sivani, [2]Dantuluri Thanusha, [3]Akella Surya Rohit, [4]Chukka Kundana Siri
MVGR College of Engineering (A), Vizianagaram – 535 005, India
Email: bsaisivani@gmail.com

Received: 26th October 2018, Accepted: 29th January 2019, Published: 28th February 2019

Abstract
Artificial Intelligence is the term often heard in the field of technology over the past few years. The vision of this paper is to provide a keen understanding about three Classification techniques for Machine Learning. They are Decision trees, Naive Bayes algorithm and Support Vector Machines. The foundation for this paper is based on several researches which were done previously on these classification techniques. The paper is presented in such a way that it illustrates the techniques used by the researchers in their papers briefly. It also re-views the previous research done on each of the classification techniques and shows a clear view of the optimal techniques. Every research was analyzed thoroughly and has been shown in contrast with respect to other researches. These researches are an extended work to the already existing techniques with newly proposed methodologies. The review says how each technique is better than the other. To have a brief understanding of comparisons each evaluation measures were depicted in tabular format.

Keywords
Decision Tree, Naïve Bayes Algorithm, Support Vector M achine.

Introduction

Artificial Intelligence is a field in computer science that involves the study of intelligent agents. Artificial Intelligence is the stimulation of human like intelligence and behavior by the machines .The programs can mimic human intelligence like reasoning, decision making, perception, visualization, voice recognition, social intelligence etc. With its vast range of applications AI began to maneuver the researchers. As an intelligent machine it must have the ability to learn from its past its experience. The program must train itself.

In decision trees the paper reviews the studies in the construction of accurate and incremental decision tree learning algorithm. Very fast decision tree (VFDT) algorithm has been proposed by the researchers along with the perks of extremely fast decision algorithm (EFDT). The paper compares their methodologies against C4.5 standard with UCI benchmark datasets, C4.5 is a decision tree generating algorithm widely. VFDTcNB has been introduced which have the ability to apply naïve Bayes classifiers in tree leaves to classify test examples without any overhead in the training phase . Also HAAT is proposed which imparts anytime property to the Hoeffding tree (VFDT), it is an incremental decision tree learner for large data sets.

Another approach for classification is by using naïve Bayes classification. It is usually employed for text classification. One of the papers deal with the clarification for the confusion encountered in the two different models in the text classification, the multi-variate Bernoulli model and multinomial model, both are probabilistic models. The multi-variate Bernoulli model does not capture the number of times the words have occurred, here probability of all the attribute values are multiplied for the calculation of probability. In multinomial model the number of occurrences of each word is captured, here the probability of the words occurred is multiplied for the calculation of probability. The research clarifies this doubt by comparing the result obtained when these two models are implemented on five datasets using "naïve Bayes assumption". The assumption is that the probability of each word occurring in a document is independent of the occurrence of other words in a document.

The next paper is an extension to the multinomial naïve Bayes .It compares to the recently proposed "transformed weight-normalized complement naive Bayes classifier "(TWCNB) with the standard multinomial naïve Bayes(MNB).TWCNB is a modified version of MNB. The paper revisits the transformation steps leading from MNB to TWCNB. Usually, MNB usually uses word frequencies as already mentioned. But here they use TFIDF scores instead of raw word frequencies as they claim it improves the performance of MNB.

Support Vector Machine (SVM) is another classification technique, commonly used to analyze high-dimensional and sparse data and recognize patterns. In the next paper the researchers classify heterogeneous (World Wide Web) gathered from computer network to support malware detection using SVM and test its efficiency in malware analysis. The data is classified into normal data denoted by "+1" and malicious data denoted by "-1". In the next paper describes new learning strategy on the problem of overlapping in imbalanced data sets. This problem increases the training difficulty but also cause over fitting which affects the traditional classification approaches. Imbalanced problem is a situation where the number of observations or instances in one class greatly outnumbers that of the other classes i.e. they are significantly not uniform. To deal with presence of imbalance and overlapping problem in learning, a modified well-known sampling method SMOTE has been suggested. It is called as SMOTE_RM, which involves fuzzy Tomek Links (FSCT) for data

 Helix E-ISSN: 2319-5592; P-ISSN: 2277-3495

cleaning, to balance the training data as well as remove noisy data and to re-class the class boundary. The author proposed new strategy named RMLSVM incorporating SVM because SVs are needed for training data and it provides removal of other samples without affecting classification. The next paper deals with the non-linearity in SVM which is incomprehensible when extracting the rules. Five rule criteria for rule extraction have been cited, they are Comprehensibility, Fidelity, Accuracy, Scalability and Generality. Here these techniques are combined with rule extraction from SVMs to deal with the non-linearity.

Literature Review

Artificial Intelligence is a broad area in computer science. It was defines in many ways [1, 2]. A large number of tools [3] have been developed by Artificial intelligence to solve many of the computer science problems. One of such tools is classification. There are many of classification algorithms [4 - 16] which can be employed as tools for Artificial intelligence. Among the many classification algorithms three classification algorithms are revisited in this paper. They are Decision trees [5, 6and 7] Naïve Bayes algorithm [8, 9] and Support Vector Machines [10, 11 and 12].

Methodology

Three supervised leaning algorithms are reviewed in this paper. They are Decision trees, Naïve Bayes and Support Vector Machine.

Decision Trees

The first supervised learning algorithm reviewed in this paper is Decision Trees. As Very Fast Decision trees are one of the most popular decision tree algorithms for mining stream data a paper on it is reviewed. The paper [5] studied the problem of constructing accurate decision tree models from data streams. Very Fast Decision Tree (VFDT) was considered. Because it is a successful algorithm for mining data streams. In this article VFDT was extended to Very Fast Decision Tree Classifier (VFDTc). It is possible to classify new information in a single scan using VFDTc. The evaluation function used here is Information Gain and it deals with numerical attributes. It need not require domain knowledge of all the possible values of a categorical attribute. But for discrete attributes it requires the counters of the number of occurrences of an observed attribute value per class. VFDT only takes into account the information about class distributions and it doesn't require attribute values. But naïve Bayes take both prior distribution of classes and conditional probabilities of attribute vales into account. Naïve Bayes deals with heterogeneous data and missing values. VFDTc was empirically evaluated by considering three dimensional analyses. They are - error rate, learning time and tree size. They analyzed the effects of two different strategies when classifying test examples: Very Fast Decision Tree classifier using Majority Class (VFDTcMC) and Very Fast Decision Tree classifiers using naive Bayes (VFDTcNB) at leaves. In this paper the datasets used are Waveform defined by 21 numerical attributes, Waveform containing 40 Attributes and LED. The two versions of waveform dataset consist of problems with three classes. The author showed that classifier's performance can be improved using classification strategies at tree leaves. On these datasets VFDTcNB consistently has out-performed VFDTcMC. The performance of VFDTcMC approximates VFDTcNB , if the number of examples increases on Waveform data. The performance of VFDTcNB is almost constant in all learning curves, independent of the number of training examples. The error rate for VFDTcNB is lowest. The learning time is lowest for VFDTcMC. The Tree size obtained for both VFDTcNB and VFDTcMC is same. By the above comparisons it can be concluded that VFDTcNB is a very competitive algorithm even in comparison against the state of the art in batch decision tree induction. It is said that the bias-variance analysis shows that VFDTcNB generates very stable predictive models concerning variations of the training set. From the experimental evaluations of VFDTc it can be clearly illustrate the advantages of using powerful classification techniques.

Very fast decision trees are further extended to EFDT in [6].[6] Introduces a novel incremental decision tree learning algorithm that is Hoeffding Any time Tree (HATT). This is statistically more efficient than the present state of the art, Hoeffding Tree (HT). They demonstrated an implementation of HATT-"Extremely Fast Decision Tree" with minor modifications to Moa implementation of HT. It obtained significantly superior prequential accuracy on the most of the large classification datasets in UCI repository. HATT produces the asymptotic batch tree in the limit. It is naturally resilient to concept drift. At a small additional computation cost it can be used as a higher accuracy replacement for HT in most scenarios. HATT is equivalent to HT except that it uses the Hoeffding bound to determine the merit of splitting. It has been observed that VFDT (Very Fat Decision Tree) takes more time to learn progressively more difficult concepts obtained by increasing the number of classes while EFDT (Extremely Fast Decision Tree) learns all of the concepts very quickly. The HT is the de facto standard for learning decision trees from streaming data, which has been used as a base for many state of the art drift learners. The proposed implementation in this paper of the HATT algorithm of EFDT achieves higher prequential accuracy than the Hoeffding Tree implementation of VFDT on many standard bench mark tasks. They evaluated the prequential error over the duration of data stream. For evaluating they plotted error for 4 different levels of complexity. This evaluation showed how EFDT learns much more rapidly than VFDT and is less affected by the complexity of the learning task, albeit incurring a modest computational overhead to do so. The data has been generated by MOA Random Tree. It has been said that the space complexity for HT and HATT is same and equivalent to O (ndvc) where d is number of attributes in data, v values per attribute and c classes. A comparison between Concept-Adapting Very Fast Decision Trees (CVFDT) and HATT has been made. CVFDT is explicitly designed for a drifting scenario where HATT is used for a stationary scenario. CVFDT's aim is to reduce prequential error for the current window in the expectation that this is the best way to respond to drift while HATTS aims to reduce prequential error over all, for a

  Helix E-ISSN: 2319-5592; P-ISSN: 2277-3495

stationary stream so that it asymptotically approaches that of a batch learner. CVFDT builds and substitutes alternate sub trees, but HATT doesn't. CVFDT always compares the top attributes while HATT compares either the current split attribute or the null split. It was concluded that even though both of them were built using split reevaluation, the circumstances where there are used, their objectives and methods are entirely different. As for HATT, it is said that it has some inbuilt tolerance to concept drift, though it is not specifically designed as a learner for drift. It also has a profound benefit for utilizing the most useful splits.

An efficient VFDT algorithm is proposed [7]. The ground condition for fast decision-tree learning (FDT) algorithm is conditional independence assumption. The time complexity of C4.5 is O(p.q^2). But the time complexity of the proposed algorithm is O(p · q). It clearly shows that this is animportant asymptotic improvement over the time complexity of state of art C4.5. FDT performs better than C4.5 on large number of UCI bench mark datasets. It performs even better and faster than C4.5 on a huge amount of text classification data sets. It is surprising that the time complexity of their Naïve Tree algorithm is as low as naive Bayes algorithm and one- level trees. Purity – based Heuristic is adopted for determining both classification performance and computational cost. Information gain is considered as heuristic. The time complexity for selecting the splitting attribute using Independent Information Gain andusing information gain in C4.5 are similar and is equal to O(p.q). Naive Tree algorithm is based on the conditional independence assumption. For the problems with huge number of attributes,Naïve Tree algorithm scales well. In the tree growing process of Naïve Tree algorithm, each candidate attribute is examined. The candidate attribute with the highest information gain will be selected as the splitting attribute. The total time taken for tree-growing process is the sum of the time for probability estimation, partition and splitting attribute selection. In training time the text classification usually involves 1000's of attributes, from which we can observe the advantage of Naïve Tree algorithm over C4.5. C4.5 and naive Bayes are compared with Naïve Tree algorithm. Because naive Bayes is one of the most practical text classification algorithms. It has very low time complexity and has substantially good accuracy as well. The metrics used for experimentation are Information gain and Entropy which are widely used as a standard heuristic. They used the 3 evaluation measures accuracy, running time and tree size. They found that the accuracy of Naïve Tree algorithm (83.37%) is very close to C4.5's (83.57%) accuracy. Naïve Tree algorithm'stree size is equal to C4.5's tree size which is 1.0. Even though they seriously violated the independence assumption which is weaker than naive Bayes independence assumption Naïve Tree algorithm performed well. Naïve Tree algorithm inherits the superiority of C4.5 in accuracy on larger datasets. As Naïve Tree algorithm has the same time complexity with naive Bayes. Naïve Tree algorithm has a clear advantage in running time when the data set is huge and has a huge number of attributes. It can clearly be concluded that Naïve Tree algorithm performs better than naive Bayes and C4.5 in accuracy, running time and tree size.

In the below table the proposed algorithms are listed against their error rate, learning time, tree size and accuracy. The values of the parameters listed in are the average of the obtained results on various data sets like Waveform, LED, Reuters, TREC and OHSUMED.

Naïve Bayes Algorithm

The second supervised learning algorithm reviewed in this paper is Naïve Bayes algorithm. By using two different first-order probabilistic models to text classification, a naïve Bayes assumption was made. While some used multi-variate Bernoulli model with no dependencies between words. Others used a uni-gram language model with integer word counts. The paper [8] aims to clarify the confusion on these models by comparing the classification performance on five text corpora. Here a document is considered as binary vector over the space of words for multi-variant Bernoulli model. This model captures word frequency information in documents. For document representation a Naïve Bayes assumption is made. The assumption is that, the probability of occurrence of each word in a document is independent of occurrence of other words in a document. When generating a component, document can be taken as a collection of multiple independent Bernoulli experiments. Here every word in the vocabulary is considered as a word event in a component. Either the absence or Presence of every word is depends only on the document class. This model explicitly includes the probability of non-occurrence words that don't appear in the document. However, this model does not include the number of times each word occurs. This model of multi- variate Bernoulli model is compared with Multinomial model. In this multinomial model a document is considered as a sequential order of word events. It is assumed that lengths of documents are independent of class. In this model a naïve Bayes assumption was made. It is similar to that of multi-variate Bernoulli model. The similar assumption is that the probability of each word event in a document is independent of words context. But this multinomial model also includes the position of word event in the document. The experiments are made on the datasets, web pages pointed by Yahoo!, Industry Sector hierarchy, Newsgroups, WebKB, Reuters. For Yahoo dataset the multinomial event model reaches a maximum of 54% accuracy at a vocabulary size of 1000 words. But the multi-variate Bernoulli model reaches only a maximum of 41% accuracy at a vocabulary size of 200 words. We observe the similar pattern is exhibited by both the models for Industrial Sector dataset. For Newsgroups dataset both the models do best at maximum vocabulary sizes. For WebKB data the multi-variate Bernoulli model has marginally higher accuracy than the multinomial. From the experiments on these datasets it was concluded that that Multi-variate Bernoulli performs well on small Vocabulary sizes, and multinomial performs well on large Vocabulary sizes with an average of 27% reduction in error over the multi-variate Bernoulli model.

The paper [9] tells how the performance of multinomial naïve Bayes can be improved using locally weighted learning. To tackle text classification problems Naive Bayes learning algorithm is frequently used. The multivariate Bernoulli event model and the multinomial event model are the two event models commonly employed. The multinomial event model is

  Helix E-ISSN: 2319-5592; P-ISSN: 2277-3495

commonly called as multinomial Naïve Bayes. In general multinomial Naïve Bayes outdo multivariate event model. However, a new algorithm which is easy to implement and has better running time has been proposed in this paper. The new algorithm proposed is Transformed Weight-Normalized Complement Naïve Bayes Classifier. Multinomial Naïve Bayes is built upon it and resembles it. The first difference in Transformed Weight-Normalized Complement Naïve Bayes Classifier is the TFIDFN transformation. TFIDF transformation takes the original word Frequency f and transforms it. It is used to in Transformed Weight-Normalized Complement Naïve Bayes Classifier to convert the count of each word in the document. It was found that the performance can improve to a great extent using normalized average vector length (TFIDFN). The second difference is that it estimates parameters of a class by using data from all classes apart that class. It was found thatTransformed Weight-Normalized Complement Naïve Bayes Classifier performance's similarly with or without normalization of the word weights. The third difference is that the one-vs-rest approach can be used with multinomial Naïve Bayes but not with Transformed Weight-Normalized Complement Naïve Bayes Classifier. The multinomial Naïve Bayes is compared with TCNB, Transformed Weight-Normalized Complement Naïve Bayes Classifier and linear Support Vector Machine (SVM). The Sequential Minimal Optimization (SMO) algorithm is used for learning SVM. The datasets used for comparison are WebKB, Industry Sector, Newsgroups and Reuters-21578. It was found that multinomial Naïve Bayes performs inferior to other algorithms, the results for TCNB are often better than that of Transformed Weight-Normalized Complement Naïve Bayes Classifier. Therefore it seems that word-weight normalization is not needed to obtain good performance using complement naive Bayes. Multinomial Naïve Bayes is improved with TFIDF transformations, TFIDFA and TFIDFN. TFIDFA is the normalized feature vector of each document to average vector length. Because of TFIDF transformations, multinomial Naïve Bayes performance improved in almost all cases. But then, TFIDFN is not very advantageous compared to Simple TFIDF. Multinomial Naïve Bayes with TFIDFA outperforms TCNB. Multinomial Naïve Bayes in conjunction with Locally Weighted Learning (LWL) on small (n=50), medium (n=500) and large (n=5000) subsets and multinomial Naïve Bayes are compared. It is found that LWL can enhance the performance of multinomial Naïve Bayes for small and medium sub sets. Multinomial Naïve Bayes with TFIDFNA and LWL and SMO with TFIDFN is compared. As there is parameter choice optimistically biased results are obtained. Though there is an optimistic bias for multinomial Naïve Bayes results, SMO performs better in almost all cases. The overall conclusion is that it is better to prefer SVM for maximum accuracy.
From these above models of naïve Bayes Classifier, we conclude that multinomial event model performs better than multi-variate Bernoulli model. But this Multinomial Naïve Bayes Produce optimistic bias results with SMO. So, it is preferable to choose SVM for better accuracy.

Support Vector Machines
The third supervised learning algorithm reviewed in this paper is Support Vector Machines. Support Vector machines are one of the tools for classification. The Support Vector machine operates on the principle that a boundary is drawn to classify the objects. All the objects inside the boundary are classified to be positive and all the other objects are classified to be negative. The boundary is called hyper plane. In two Dimensional Spaces Hyper plane a line is drawn to divide the plane into two subspaces where each class rest in any of the subspaces. They operate by construction of marginal hyper planes. SVM's are mostly used to classify the Linear Data. SVM can be used in prior to NBC and Decision Trees because the redundancy in SVM is low. Minor changes in the data cannot affect the hyper plane. At initial stages svm were designed to solve pattern recognition problems. Recently Svm has been fused with alternative loss function to solve regression problems. When the data is to be classified according to boundary conditions decision tree cannot be construct with all attributes, which enables us to lose at least 30% of data. Considering the efficiency and speed with linear data SVM is mostly preferred.
In the paper [10] the author identifies the malware sent through various ip addresses. It is done using one of the classification techniques Support Vector Machine (SVM). The true aim of the author is to classify heterogeneous datasets comprising of various ip addresses which contain malware. The system was trained such that any other ip address belonging to the particular domain can be classified as malicious addresses. In the classification positive class was denoted with '+1' and negative class was denoted by '-1'. Then the classification is started with the theoretical explanation of SVM and the equations representing the Formation of an SVM. In this process a problem of non-linearity of the data is occurred. Since SVM does not support computations on non-linear data. The use of a kernel function is suggested. A kernel Function is a function which defines similarity between two objects. The standard data set N6 is used throughout the paper. For computing the kernel matrix each and every element is considered and similarity between them is calculated. The techniques used for evaluating the proposed algorithm are explained. The kernel function parameters were considered as gamma=0,5, r=1 and d=3. Experimentation results are specified by considering three different methods and data-sets. The preliminary evaluation measures Accuracy, Sensitivity, Specificity and AUC-measure, F-measure and Precision were calculated. The first method illustrated was Cross validation with five number of folds. The values obtained for the evaluation measures were as follows Accuracy (0.9398), Sensitivity was (0.9808), Specificity (0.8613), Area under ROC curve being (0.9937), F-measure (0.9552) and precision (0.9309). The second method considered was Leave-One-Out method which scored Accuracy (0.9422), Sensitivity (0.9732), Specificity (0.8832), Area under ROC curve is (0.9927), F-measure is 0.9927 and Precision is 0.9407. The final method considered is Random Sampling with five repetition of training which scored the following results Accuracy (0.9937), Sensitivity (0.9774), Area under ROC Curve (0.9901), F-

  Helix E-ISSN: 2319-5592; P-ISSN: 2277-3495

measure (0.9623) and Precision is (0.9475). From the above results it can be illustrated that Random Sampling with repetition of training is the most prominent technique.

SVM's are mostly used to classify the Linear Data. In the paper [11] the author starts with a statement that Support Vector machines are tools for classification techniques. It was illustrated that the main advantage of SVM is that they are efficient when working with Linear Data-Models. However, they are inefficient in the case of Non-Linear Datasets. It was stated that the SVM's can be further developed by extracting rules from them. Five criterions to evaluate the rule extraction algorithm cited are comprehensibility, fidelity, accuracy, scalability and generality. Comprehensibility is the extent up to which the rules can be understood by humans. Fidelity is the extent which the rules pose a similarity from the SV they were extracted. Accuracy is how correct the predictions can be obtained from the rules. Scalability is the extent to which the algorithm supports I.e., the algorithm must be working for large data-sets. Generality is the extent to which the restrictions are present on model architecture. It is stated that with increase in data set size the efficiency of traditional SVM's decreases. So SVM is combined with feature selection. This can yield better results when compare to traditional SVM alone. The pedagogical rule extraction method is quoted in the paper. The ideology of pedagogical method is "learn what the SVM have learned". In the phase of Rule Extraction from SVMs the rule induction algorithms are used. The rule induction algorithm state that the decision tree is formed on basis of a certain factor. The factor is Gini Index. The attribute with minimum Gini index is considered to be the splitting attribute. In the next phase the essence of pedagogical method is explained. In the pedagogical method the original data is changed into SVM predictions and the labels were changed. Then the decision tree is applied on the transformed dataset. Then a brief introduction on RE-RX pedagogical algorithm (recursive algorithm) is given. It is a technique that implies the splitting of the input space into sub-spaces. In the final partition there will be only continuous attributes. The difference to be noted is, with naive decision tree construction all the splitting of the nodes is done only through single discrete attributes. But in this model the splitting is done through a combination of continuous attributes. In the next phase when a rule was being extracted from SVM with feature section a problem was faced. The problem faced was the time consumption of the algorithm when applied on large data sets. In order to reduce the time for analysis it was suggested to remove all the unnecessary and redundant features. The algorithm was applied on the several data-sets and the results were evaluated using fidelity and accuracy. When no rules were considered the support, fidelity and accuracy for C4.5 was (0.0.64, 0.658, 0.688) and the algorithm scored (0.1232, 0.821, 0.659). When 1 rule was considered the support, fidelity and accuracy for C4.5 was (0.664, 0.886, 0.814) and for the algorithm was (0.489, 0.937, 0.84).When CART data-set with no rules were applied the support ,fidelity and accuracy for C4.5 are (0.132,0.54,0.555) and algorithm are (0.069,0.832,0.645).When CART technique with 1 rule the support, fidelity and accuracy were (0.151,0.909.0.865) and for algorithm with CART were (0.148,0.703,0.723). From the above results it can be clearly identified that the rule 0 (with feature selection) yields better results than native decision tree techniques. The CART technique is better when compared to native C4.5 technique. As CART technique involves both classification and regression the data-set which is to be analyzed or classified goes on with two stages of refinement. On a whole in this paper the author aims to introduce a new technique to improve the efficiency of SVM and to overcome the limitations of SVM. The entire Flow of the paper proceeds to explain the class-imbalance problem and suggesting a new method that can reduce the limitations of the naive SVM and make the data-set more better to get efficient results.

In the paper [12] the author wants to deal with the problem of imbalanced and overlapped data-sets. An algorithm was developed to extract the informative support vectors for system training. New clusters are formed by cleaning the data. An example of two-class Classification problem was quoted. The training sets belonging to one class are relatively huge when compared to another. It states that imbalanced data-sets may reduce the performance of the classifier. So, Synthetic Minority Over-Sampling Technique was used. The technique is to remove noisy examples and re-define the class-boundary data to get better generalization. Local SV was used to keep the coherent characters in the distribution. Wilson's Editing technique was used to remove only majority class instances which can contain noise and those classes are removed by K-NN algorithm with k value being 3 (k=3).The dataset considered was not transformed into a kernel matrix because of time and scalability issues. The following algorithm was described. The nearby points of the samples which were classified to be positive were considered. Then by using Synthetic Minority Over-Sampling Technique, new synthetics were generated. The SMOTE_RM method has very less effect on algorithmic complexity of the considered point's neighbors. Redefining the boundaries and the cleaning of data helps to reduce the size of the actual larger class-bound. SV's can be safely utilized and if required they can be safely removed. The classes were classified and evaluated. "Reserved Regions" were constructed from the purely balanced classes. The data Processed was evaluated by G-mean and F-measure. In The Abalone dataset the ratio of imbalance is 40:1 and g-mean was 0.7859 and f-measure was 0.3138. In the MCD data-set the ratio of imbalance is 100:1 and g-mean is 0.5207 and f-measure is 0.0689. It was clearly shown that more the imbalance lesser the possibility for efficient classification. Through this paper the author wants to states that using SVM and few other data processing techniques, the imbalanced classed-data-sets can be converted to a balanced data-set. They can also be evaluated to increase the accuracy and precision.

Conclusion

The conclusion of is that VFDTcNB and VFDTcMC are same and VFDTcNB generates a stable predictive decision tree. CVFDT and HATT were built based on the split revaluation, but their objectives and methods were different. As NT inherits the superiority of C4.5, it performs better than Naïve Bayes and C4.5 in accuracy, running time and tree size. Decision trees don't work well with continuous target variable, so Naïve Bayes classifier was introduced. Multinomial

  Helix E-ISSN: 2319-5592; P-ISSN: 2277-3495

Naïve Bayes model performs much efficient than multi-variate Naïve Bayes model. But MNB produces optimistic bias results with SMO. So, SVM is preferable for better accuracy. SVM is good as compared to other classifiers as the computational complexity is reduced and classification efficiency is increased when compared to any other nonlinear classifier. SVM identifies the malware sent through various IP addresses through the Random Sampling with repetition of training technique. In order to improve the efficiency and to overcome the limitations of SVM, CART technique was introduced. Through this SVM and other few processing techniques, the imbalanced classed data-sets can be converted to balance data-set and can be evaluated to increase the accuracy and precisions.

Future Work

In the future we would like to review other more efficient classifiers like Artificial Neural Networks, Kernel methods, Gaussian mixture model and k-nearest neighbor algorithm.

References

1. Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press; 1992.
2. Hewitt C, Bishop P, Steiger R. Session 8 formalisms for artificial intelligence a universal modular actor formalism for artificial intelligence. InAdvance Papers of the Conference 1973 (Vol. 3, p. 235). Stanford Research Institute.
3. Pham DT, Pham PT. Artificial intelligence in engineering. International Journal of Machine Tools and Manufacture. 1999 Jun 1; 39(6):937-49.
4. Ruggieri S. Efficient C4. 5 [classification algorithm]. IEEE transactions on knowledge and data engineering. 2002 Mar; 14(2):438-44.
5. Joao Gama, Ricardo Rocha, Pedro Medas. Accurate decision trees for mining high-speed data streams. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining 2003 Aug 24 (pp. 523-528). ACM.
6. Chaitanya Manapragada, Mahsa Salehi Geoffrey I .Webb. Extremely Fast Decision Tree. arXiv preprint arXiv:1802.08780. 2018 Feb 24.
7. Su J, Zhang H. A fast decision tree learning algorithm. In AAAI 2006 Jul 16 (Vol. 6, pp. 500-505).
8. Andrew McCallum and Kamal Nigam. A comparison of event models for Naive Bayes text classification. InAAAI-98 workshop on learning for text categorization 1998 Jul 26 (Vol. 752, No. 1, pp. 41-48).
9. Ashraf M. Kibriya, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes .Multinomial Naive Bayes for text categorization revisited. In Australasian Joint Conference on Artificial Intelligence 2004 Dec 4 (pp. 488-499). Springer, Berlin, Heidelberg.Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999).
10. Michał Kruczkowski, Ewa Niewiadomska-Szynkiewicz. Support vector machine for malware analysis and classification. In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 02 2014 Aug 11 (pp. 415-420). IEEE Computer Society.
11. Si Xiao Yang, Ying Jie Tian, Chun Hua Zhang. Rule extraction from support vector machines and its applications. In Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03 2011 Aug 22 (pp. 221-224). IEEE Computer Society.
12. Tong Liu, V Liang, Weijian Ni2. A learning strategy for highly imbalanced classification. In Proceedings of the Third International Conference on Internet Multimedia Computing and Service 2011 Aug 5 (pp. 116-119). ACM.
13. Yegnanarayana B. Artificial neural networks. PHI Learning Pvt. Ltd.; 2009 Jan 14.
14. Shawe-Taylor J, Cristianini N. Kernel methods for pattern analysis. Cambridge university press; 2004 Jun 28.
15. Reynolds D. Gaussian mixture models. Encyclopedia of biometrics. 2015:827-32.
16. Jiang S, Pang G, Wu M, Kuang L. An improved K-nearest-neighbor algorithm for text categorization. Expert Systems with Applications. 2012 Jan 1; 39(1):1503-9.

  Helix E-ISSN: 2319-5592; P-ISSN: 2277-3495