DOI: 10.21917/ijsc.2019.0276

IT HELP DESK INCIDENT CLASSIFICATION USING CLASSIFIER ENSEMBLES

S.P. Paramesh and K.S. Shreedhara
Department of Computer Science and Engineering, University B.D.T College of Engineering, India

Abstract

Proper assignment of IT incident tickets raised by the end users is a very crucial step in an IT Service management system. Incorrect manual selection of incident category while raising the ticket causes assignment of incident to a wrong domain expert team which in turn results in unnecessary resolution delay and resource utilization. In this work, we proposed machine learning based model for auto categorization of incident category by mining the user's natural language description of the incident. Classification techniques such as Naive Bayes and Support Vector Machines are used as base classifiers to model the incident classifier system. To further analyse the classifier performance we used the ensemble classifier techniques such as Bagging and Boosting to build the incident classifier model. The performance of base classifiers and ensemble of classifiers are analysed using various performance metrics. Ensemble of classifiers outperformed well in comparison with the corresponding base classifiers. Pre-processing of the IT incidents description data is one of the key challenges in this research work due to its unstructured nature. The proposed automated incident classification model results in simplified user interface, faster resolution time, improved productivity and user satisfaction and uninterrupted flow in business operations. The real world IT infrastructure incidents data from a reputed enterprise is used for our research purpose.

Keywords:

Machine Learning, Incident Classification, Ensemble Classifiers, Naive Bayes, Support Vector Machine

1. INTRODUCTION

Most of the current IT organizations use the Information Technology Service Management (ITSM) frameworks to manage the IT operations and services [1]. The ITSM framework helps in improving the organization business by providing a high quality of services to the end users or customers. Incident management is an element of ITSM deals with managing the IT incidents from submission of tickets till the closure. An incident is defined as "An unplanned interruption to an IT service or reduction in the quality of an IT service" [2]. IT Helpdesk systems are the place and point of contact for the end user to raise the incident tickets and to get the resolution for the same. The incidents raised by the end users of the organization should be handled by the support team as early as possible for the normal functioning of the business. In a typical IT organization, employee's experiences lot of issues with respect to facilities, infrastructure, applications, HR related issues, travel etc. The employees of the organization usually raise the issue ticket using IT helpdesk portal which is typically of web based. The submitted incidents will then be assigned to a proper resolver group comprising of domain experts team specialized in the particular area.

User submits the problem ticket by selecting the incident category, sub category and by entering the incident description. The user also selects various other fields like incident priority, severity and attaches the supporting file if any for quicker response. Manually selecting the incident category by end user may result in forwarding the tickets to wrong resolver group as it is completely based on the user's knowledge and understanding in selecting the right category of incident ticket. Improper selection of incident category further leads to reassignment of tickets, dragging the resolution process, unnecessary use of domain resources effort and time, user satisfaction deterioration and ultimately impact the normal functioning of the business.

To overcome all these problems, we can automate the manual incident ticket classification process by using Machine Learning (ML) algorithms [3] [4]. ML algorithms are used to build an automated incident classifier system which auto categorises the tickets into one of the predefined category by parsing the unstructured incident description provided by the helpdesk user. Classifier models can be developed by using both supervised and unsupervised machine learning methods. When the label or category of the historical training ticket data is known in prior then supervised machine learning techniques like classification algorithms can be used to build classifier models [5]-[8]. Unsupervised machine learning techniques like clustering can also be used to group similar kind of tickets to a particular category followed by labelling of the clusters when the ticket category is unknown [9].

In this research paper, we used multiple classifier models or classifier ensembles techniques to analyse the accuracy of the incident classification system. Ensemble involves building the multiple classifier models using different sub samples of the training data and then combining the results using averaging or majority voting techniques. The result of combining different models gives better accuracy when compared to the individual models [16]. The ensemble of classifiers also reduces variance which helps to avoid over fitting problems on the test data.

In this research work, Naive Bayes (NB) and Support Vector Machines (SVM) are used as a baseline classifiers to build the incident classifier followed by building the classifier models using ensemble methods [17][18]. Popular ensemble methods such as Bagging and Boosting techniques are applied on the base classifiers for the incident categorization purpose.

Bagging is an ensemble technique which involves building the multiple models of the same type in parallel using the random sub samples of the original data set and then combining the predictions of these models using averaging or majority voting technique. Each of the subsample is selected based on the random sampling with replacement method. Bagging is also called as bootstrap aggregation [16]. Multiple Naive Bayes ensemble called as Bagging-NB and multiple Support Vector Machine ensembles called Bagging-SVM models are used as a part of bagging ensemble techniques in this research work.

Boosting is an ensemble technique that involves creating a sequence of models from the random samples of the training data which tries to rectify the mistakes of the previous predictor classifiers in the sequence. If an instance was misclassified, it tries to increase the weight of this instance and the misclassified instances were given more weightage in the next model.

Generally, the boosting techniques are used to boost the performance of weak learners such as decision trees, neural network etc. In this research work, we used the boosting algorithm called Adaboost (Adaptive boosting) model to build the incident classifier [17] [18]. SVM based boosting technique called Adaboost-SVM is used as a part of boosting ensemble technique [19].

The performance of all the ensemble models are evaluated and compared against individual baseline classifier models using various classification performance metrics such as Accuracy, Precision, Recall and F-score. The incident classifier model which outperforms well on the test data when compared to all other chosen classifier models is used as a predictive model to categorize the new unlabelled incident.

A real world IT infrastructure incident data of a reputed enterprise is used for our research purposes. Typical IT infrastructure problems can be related to hardware issues, software issues, network issues, email issues etc. The structure of the typical IT infrastructure incident tickets raised by end users is given in Table.1.

Table.1. Typical IT infrastructure helpdesk ticket data

ID	Incident Category	Priority	Submitter	Description	Status
300	Network issue	High	XXXX	Unable to connect to LAN	Open
301	Hardware Problem	High	MNOP	Hard disk crashed	Closed
302	VPN issue	Medium	YYYY	VPN is not working	Assigned
303	Outlook	Medium	QRST	Please configure Outlook	In progress

The proposed classifier model uses the historical incident data containing description about particular incident and its corresponding label for training the system. The incident classification can be considered as an instance of document categorization in which each incident description is assumed to be a text document and its corresponding category as the document label. Main objectives of this paper are:

- Handling the incident data related issues for the chosen IT infrastructure incident dataset.
- Modelling the incident ticket classifiers using Ensemble of classifier techniques such as Bagging and Boosting ensemble.
- Evaluation and comparison of various ensemble classifiers with traditional base classifiers using various classification performance evaluation metrics.

The advantages of developing such an automated incident ticket classifier system includes simplified web user interface of the incident management tool, quicker resolution, effective use of domain resources used to resolve the issues, improvement in customer satisfaction and as a result at the end of the day there is an improvement in business growth.

2. RELATED WORKS

In the literature, very limited number of research works carried out in the context of automation of IT helpdesk incident ticket classification. Some of the prior works in this area are detailed below.

Gupta et al. [3] proposed a method for routing the incident ticket to correct subject matter expert teams to resolve the tickets. The method correlates the incoming incident with configuration items like systems, software’s etc. stored in configuration management database. The unstructured ticket description and other structured ticket fields are used to identify the relevant configuration items. SVM models are used to classify the incoming ticket to a particular category based on the keywords.

Mucahit et al. [4] developed an enhanced issue tracking system based on machine learning which auto routes the ticket to relevant person for ticket resolution. The proposed model uses the bag of word approach to convert the ticket descriptions into a feature vector representation form. The term weighting approaches like binary, term-frequency (tf) and term frequency-inverse document frequency (tf-idf) were used to represent the features. The classification algorithms like decision trees, SVM, K-Nearest Neighbour and Naive Bayes are used for modelling the ticket classifier. Experimental Results indicates that accuracy of the classifier depends on the training data, weighting method and classification technique used for building the model.

Agarwal et al. [5] discusses about building a cognitive IT support system that does automatic problem diagnosis by identifying the problem category, analyses the root cause of the problem and provides automatic resolution by mining the historical problem tickets descriptions and associated resolutions. Machine learning and Natural language processing techniques are used to develop such system.

Silva et al. [6] uses Support Vector machines (SVM) to automate the IT incident management process. The proposed SVM based incident classifier is used for one of the real world incident ticket and the model achieved an accuracy of 89% on the test data.

Paramesh et al. [7] proposed a method for building an automated IT service desk ticket classifier system by using traditional supervised machine learning techniques like Logistic regression, K-NN, Multinomial Naive Bayes and SVM. Methods to handle data related challenges found in the chosen dataset are discussed in detail. For the chosen dataset, SVM based ticket classifier outperformed well when compared to all other models.

Al-Hawari et al. [8] developed a methodology based on machine learning for accurate ticket classification in IT helpdesk for German Jordanian University. Along with auto categorization of service desk tickets, the proposed system also offers administrator view to manage tickets and user view to report issues and request IT management services. The model uses Support vector machines to build the service desk ticket classifier system.

Roy et al. [9] proposes an incident classifier model based on the unsupervised machine learning techniques like clustering to cluster the incident tickets using the prior user’s ticket description. The proposed method uses k-means clustering based on a new distance metric which uses the combination of Jaccard distance and cosine distance for fixed and free fields of the tickets respectively. The clustering is then followed by labelling by extracting logical item sets for each of the clusters.

We also did the literature survey in the field of text mining since our research intent is a classical use case of a text classification problem. Some of the previous research works in the field of text classification are discussed below.

Ikonomakis et al. [12] discusses the various components involved in text classification process. The different methods and techniques required to achieve pre-processing of data, Feature vector representation of data, feature selection, feature extraction and to build and evaluate the classification models are discussed in detail.

Allahyari et al. [13] give an excellent review on the various machine learning techniques used for the text mining problem. The paper discusses various data pre-processing techniques, classification and clustering methods to achieve the text mining.

Mironczuk et al. [14] provides overview the state-of-the-art elements of text classification and their associated techniques. The paper discusses various text classification elements like data acquisition, data analysis, Feature vector construction for selected features followed by training the classification model and model evaluation.

Kowsari et al. [15] provide an excellent review on various feature extraction and dimensionality reduction techniques, various text classification and model evaluation methods. The paper also discusses the limitations of each text classification technique and their application in the real-world.

Our research work uses the Ensemble of classifiers techniques. In the literature, different types of ensemble techniques were discussed in the context of classification problems.

Breiman [16] discusses the Bagging ensemble technique which builds multiple models of the predictor and then combines these predictions to further enhance the accuracy of the predictor. Bagging of classification and regression trees were implemented on various datasets for the experimental purposes.

Review of different ensemble techniques such as Bayesian averaging, error correcting output coding, Bagging and Boosting were discussed in [17]. The work explains why the performance of ensemble classifiers is better when compared to individual base classifiers models.

Dong et al. [18] discusses the comparison of different ensemble techniques like bagging and boosting used in the text categorization by considering SVM and naive bayes as the base classifiers.

Sharma et al. [19] uses the boosting ensemble technique applied to the base SVM classifier for the sentiment based classification of online text. The performance results shows that the boosting ensemble applied to SVM outperforms well when compared to the performance of individual base SVM.

3. PROPOSED METHODOLOGY

Incident categorization is an instance of document classification problem in which each incident description refers to the document and the incident category as the label of the document. So our proposed incident ticket classifier model involves all the phases of a typical text classification problem such as data acquisition, analysis of raw dataset, data pre-processing, feature vector representation, proper feature selection followed by model building and evaluation using performance metrics [12] [14]. To develop an incident classifier system, we used historical incident dataset containing incident description and its category as the mandatory fields. The various other structured fields of the historical incident ticket data such as submitter, priority, severity, date and time of ticket creation etc. were ignored since they do not contribute in building the proposed ticket classifier models. The high level design of the proposed incident classification system is shown in Fig.1.

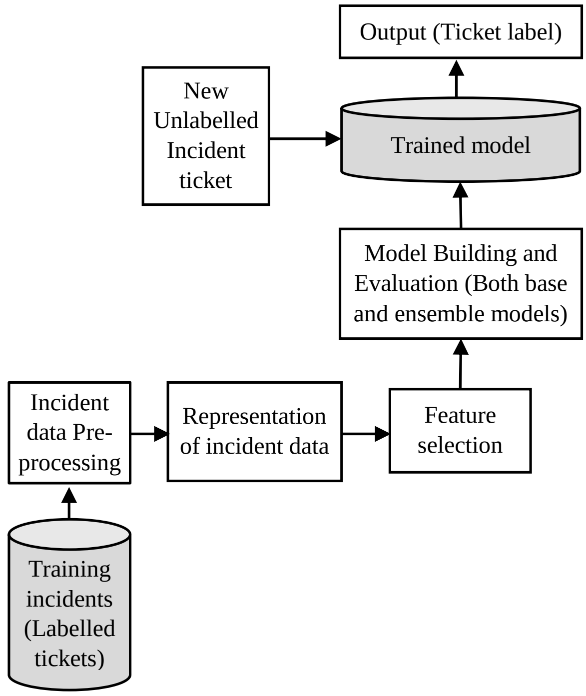

Fig.1. Proposed high level design diagram of the IT Helpdesk incident classification

The main components of the proposed IT incident classifier system shown in Fig.1 are explained as follows.

3.1 DATA PRE-PROCESSING

Pre-processing being one of the most important phases of the data mining process usually involves cleaning of the raw data. In this work, the historical ticket data containing the unstructured ticket description is pre-processed to remove any unwanted and noisy data. The IT incidents description data chosen for this research work had huge amount of unwanted data like:

- Stop words
- Special characters
- Features like date and time.
- The functional words like determiners, conjunctions, prepositions, pronouns, auxiliary verbs etc.
- User details like name, phone numbers, and email address were also present in the incident description.

The pre-processing block cleans all such undesired data since they do not aid in incident ticket classification.

Standard English stop word list is used to remove the commonly used stop words from the ticket data. Appropriate regular expressions or pattern recognizers were developed to remove entities like date, time, user's name, phone numbers and email address if any present in the incident ticket description. Parts of Speech (PoS) tagging is done to each word to eliminate the functional words from the ticket data. Stemming also performed as part of pre-processing which reduces the words to its base form. Standard porter stemmer algorithm is used for this purpose.

The performance of the classifier models depends on how clean is the training data used to model the classifier. So careful analysis and proper incident data pre- processing improves the accuracy and model building time of the classifier.

3.2 INCIDENT TICKET DATA REPRESENTATION USING FEATURE VECTOR

Pre-processed training data containing historical incident descriptions must be represented in numerical form before applying any machine learning algorithms. A Feature vector is constructed for each incident ticket description using bag of words (BoW) approach or vector space model [13]. In this representation, each vector element corresponds to a unique feature taken from the entire corpus of documents i.e., in our case it is the training incidents description data. Each feature is assigned a numeric value using Term Frequency-Inverse document frequency (TF-IDF) weighting scheme which represents the importance of the word in a particular document. Mathematically, tf-idf of a particular term t in a given document d is given below in Eq.(1)

$$tf\ idf = tf\left(t,d\right)\cdot\log\left(\frac{n_d}{n_d\left(t\right)}\right) \tag{1}$$

where, $tf(t,d)$ is the frequency of the term t in the document d. $n(d)$ and $n_d(t)$ respectively represents the total number of documents and the number of documents containing the term t.

3.3 FEATURE SELECTION

Generally, not all the features of the training data will contribute in the mining process. Unnecessary usage of features to build the classifier model results in consumption of more space and reduces the performance of the classifier model in terms of accuracy and time. So it is necessary to extract only relevant attributes for modelling the ticket classifier system. In this research work, we extracted the number of relevant features necessary to build the ticket classifier model using chi-square method [12].

Chi-square (χ^2) metric is used to measure the independence between the term t and the class c. Mathematically, Chi-square (χ^2) metric is defined as below in Eq.(2),

$$\chi^2\left(t,c\right) = \frac{N\left(AD-CD\right)^2}{\left(A+C\right)\left(B+D\right)\left(A+B\right)\left(C+D\right)} \tag{2}$$

where, N is the total number of documents in the training data, A is the number of documents in class c containing the term t, B is the number of documents containing term t in other classes, C is the number of documents in class c that do not contain the term t and D is the number of documents that do not contain the term t in other classes.

Once the Chi-Square (χ^2) value for all the features is obtained, only those features having a value greater than the specified threshold value are selected and are used for training the ticket classifier model.

3.4 MODELLING AND EVALUATION

In this work, individual base classifier models are built by using Naive Bayes and SVM models.

3.4.1 Naive Bayes (NB):

Naive Bayes is the simplest probabilistic classifier based on the Bayes rule [14] [15]. It is widely used in text document classification problems. Naive Bayes assumes that words or features of a document are independent of each other. High level description of the Naive Bayes model for the document classification (i.e., in our case it is incident classification based on description) is described below in Eq.(3)

$$P\left(c\middle|d\right) = \frac{P\left(d\middle|c\right)P\left(c\right)}{P\left(d\right)} \tag{3}$$

where, $P(c|d)$ represents the posterior probability of class c given the document d. $P(d|c)$ is the conditional probability of document d belonging to class c and can be calculated as the product from the likelihoods of the individual words or features of the document d. $P(c)$ is the prior probability of each class. $P(d)$ is the evidence and it is equal for all classes and can be ignored.

Text classifier based on Naive Bayes assumes distributions of different terms are independent from each other and hence the Naive Bayes algorithm can be rewritten as in Eq.(4)

$$P\left(c\middle|d\right) = \frac{P\left(c\right)\prod_{w\in d} P\left(d\middle|c\right)^{n_{wd}}}{P\left(d\right)} \tag{4}$$

where n_{wd} is the number of times the word w occurs in the document d and $P(w|c)$ is the probability of word w given the class c and can be calculated using below Eq.(5)

$$P\left(w\middle|c\right) = \frac{1+\sum_{d\in D_c} n_{wd}}{k+\sum_{w'}\sum_{d\in D_c} n_{w'd}} \tag{5}$$

where k is the number of distinct classes in the training dataset.

Naive Bayes ticket classifier model finds the incident ticket category c of the incoming incident containing the description d by finding the posterior probabilities for each class c. The class with the highest posterior probability is assigned as the ticket category of the new unlabelled incident ticket.

Our research work is a multiclass problem and uses the Multinomial Naive Bayes (MNB) model which assumes feature probabilities are independent of each other and considers the frequency of each word in a document for the classification.

3.4.2 Support Vector Machines (SVM):

The SVM algorithm was first developed by Vapnik and Chervonenkis [10]. SVM algorithms are first applied to text classification problems by Joakins in 1998 [11]. SVM is basically used for binary classification problem in which the given data point belongs to either positive or negative class. In the context of document classification when the documents are represented as a data points in a high dimension space, there exists many hyper planes that separates the data points into positive and negative instances. SVM algorithm tries to find the optimum hyper plane with the maximum margin ξ from positive and negative instances [13]. The documents with distance ξ from the hyper plane are called support vectors. Since our research problem is an instance of multiclass document classification, we model such a multiclass SVM by integrating the outputs of different binary classifiers based on one versus all method [20].

To further analyse the performance of the incident ticket classifier model, ensemble techniques such as Bagging and boosting are applied on the chosen baseline classifier models. In this research work, below ensemble models are used to build the incident classifier models.

3.4.3 Bagging Naive Bayes (Bagging-NB):

Bagging-NB is an ensemble classifier based on bagging technique. Different Naive Bayes predictor models are generated in parallel using random sub samples chosen with replacement out of the training data. The results of the predictions are then combined using the majority voting or averaging technique to obtain the accuracy of the composite Bagged Naive Bayes (Bagging-NB) model.

3.4.4 Bagging Support Vector Machines (Bagging-SVM):

Multiple Support vector machines classifiers are modelled using different random sub samples of the training data chosen with replacement. The results of the predictions are then aggregated to obtain the final accuracy of the Bagged Support Vector Machines (Bagging-SVM) model.

3.4.5 AdaBoost- SVM:

Adaboost is an ensemble predictor model based on the boosting technique. The steps involved in building the Adaboost classifier model are given below.

Step 1: Initially, all instances of the training data are assigned equal weights.

Step 2: A model is built using a random subset of the training data sampled with replacement.

Step 3: The generated model is then used to make predictions on the entire dataset.

Step 4: While building the next model, more weightage is given to the incorrectly classified instances i.e. weak learners are created at each step sequentially.

The process is repeated until a pre-defined number of weak learners are reached or the error function does not change.

In this work, we applied the Adaboost algorithm on the SVM base classifier called as Adaboost-SVM to build the incident classifier model [19].

Once all the models are generated, the performance of both base classifier models and various ensemble classifier models on the test data is evaluated and compared using performance metrics such as Accuracy, Precision, Recall and F-score. The performance metrics are mathematically defined as follows.

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \tag{6}$$

$$Precision = TP / (TP + FP) \tag{7}$$

$$Recall = TP / (TP + FN) \tag{8}$$

$$F\text{-}score = 2\times(Precision\times Recall) / (Precision + Recall) \tag{9}$$

Here, classifier performance metrics are defined using *TP* (True Positives), *TN* (True Negatives), *FP* (False Positives) and *FN* (False Negatives) of the classification results. Accuracy is the number of correct predictions out of all the classifier predictions. Precision is the ratio of correctly classified positive instances to all the positives. Recall is the fraction of known positives that are correctly classified. F-score is the weighted average of precision and recall.

3.5 TRAINED MODEL

After evaluating and comparing the performance of base classifiers and ensemble classifiers using various classifier metrics, the best predictive model which performs well on the test dataset is selected as a trained model and is used to automatically categorize the new unlabelled incident.

The algorithm for the proposed IT incident ticket classifier model is given below.

3.6 ALGORITHM: IT INCIDENT TICKET CLASSIFIER

Input: TR is the set of labelled IT incident ticket descriptions used for training the model, TE is the set of unlabelled IT incidents descriptions used for testing the model.

Output: Predicted ticket category for instances in test set TE.

Step 1: Initial analysis of the unstructured training data TR for the presence of any class imbalance issues, unwanted features and any other noisy data.

Step 2: **for** each of the incident ticket description in TR and TE **do**

- **a.** Extract all the words or features of incident description using tokenization.
- **b.** Remove the stop words from the tokenized words.
- **c.** Remove the special characters, features like date and time using the appropriate pattern recognizers.
- **d.** Remove all the functional words using Parts of Speech (POS) tagging.
- **e.** Remove the entities like user name, phone numbers and email ids using the appropriate pattern recognizers.
- **f.** Perform the stemming of words using porter stemmer.

end for

Step 3: Using the pre-processed incident descriptions in TR, construct the Feature vector representation for each ticket instance based on the Bag of words model.

Step 4: Reduce the feature set using the chi-square (χ^2) metric as a part of dimension reduction.

Step 5: Build the classifier model using the chosen base classifiers and ensemble of classifier models.

Step 6: Evaluate the different classifiers performance using various performance evaluation metrics and the best performed model is chosen as the predictive model.

Step 7: **for** each of the incident description in TE **do**

a. Use the predictive model to find the label of the

b. Unlabelled incident ticket.

end for

4. RESULTS AND DISCUSSIONS

The research findings at different phases of the incident categorization process are discussed below.

4.1 INCIDENT DATASET ANALYSIS AND DATA PRE-PROCESSING

Already resolved historical IT infrastructure incident ticket data of a reputed enterprise is used for our research purpose. A typical IT infrastructure data generally contains categories related to hardware problems, OS problems, network issues, software related issues, VPN issues etc.

Initial analysis of the data revealed the following details.

- Total number of tickets collected: 10742 instances.
- No of distinct classes present in the dataset: 18 distinct classes.

Analysis of the training data reveals that our chosen data set had multiple incident categories and hence it is multi-class problem. The view of the different incident categories in the training data and tickets distribution across these multiple categories is given in Fig.2.

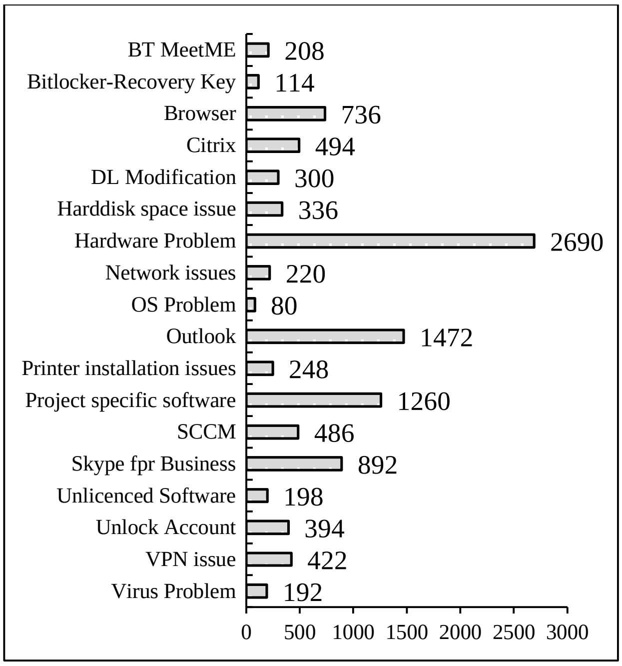

Fig.2. IT Incident ticket distribution across different classes

It indicates from Fig.2 that some of the classes contain more number of incidents and some with very less number of instances. Class imbalance problems may affect the classifier accuracy and hence techniques like random under sampling and over sampling may be used to overcome this problem.

Initial analysis also revealed that the raw dataset had huge amount of unwanted features. The details of the features or words present in the training incidents before and after performing data pre-processing step are detailed below and are depicted in Fig.3.

- Total number of features before data pre-processing: 12320
- Number of words after removing the stop words: 9232

Number of unique words after removing all the unwanted features like special characters, functional words, names, email ids, phone numbers etc.: 3928.

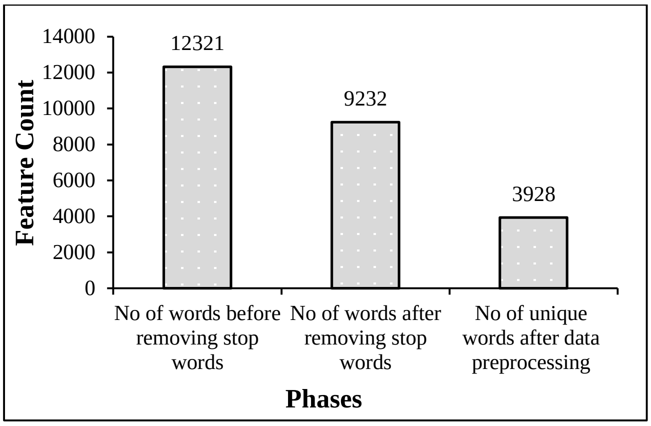

Fig.3. Summary of Features count at various phases

After the pre-processing of the incident data, a specified number of unique features (threshold =1000 features) are selected as a part of feature selection using Chi-square metric.

4.2 MODEL BUILDING AND EVALUATION

To build the different incident classifier models, the original IT incident data would be split into training and test sets with 70% of tickets (7519 tickets) used for training and rest (3223 tickets) for validating the classifier model.

4.2.1 Modelling using Base Classifiers:

In our research work, we used Naive Bayes and SVM as the baseline incident ticket classifier model. The average accuracy performance of Naive Bayes (Multinomial) and SVM using k-cross validation (k = 10) on the training data is evaluated and the results are shown in Fig.4.

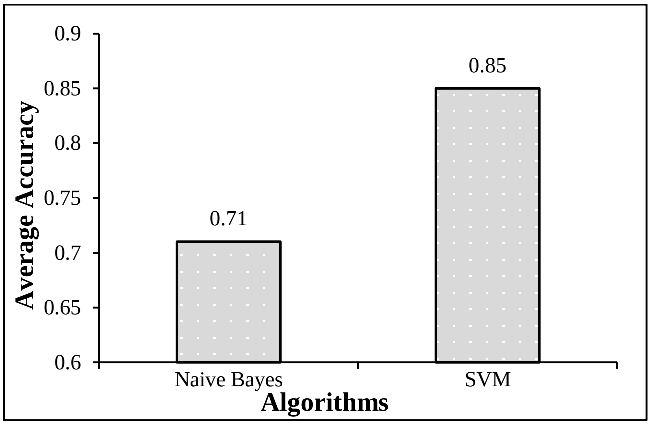

Fig.4. Average accuracy performance comparison of base classifiers using K-cross validation

The K-cross validation results shown in Fig.4 indicate that the SVM classifier model with 85% average accuracy outperformed well when compared to Naive Bayes (71%) classification model.

The performance of each base classifier models are then evaluated against the test data using various classifier performance metrics and the results are shown in Table.2 and Fig.5.

Table.2. Performance of Naive Bayes and SVM on test set

Base Classifier	Accuracy	Precision	Recall	F-score
Naive Bayes	0.7162	0.7487	0.7162	0.6754
SVM	0.8574	0.8571	0.8574	0.8497

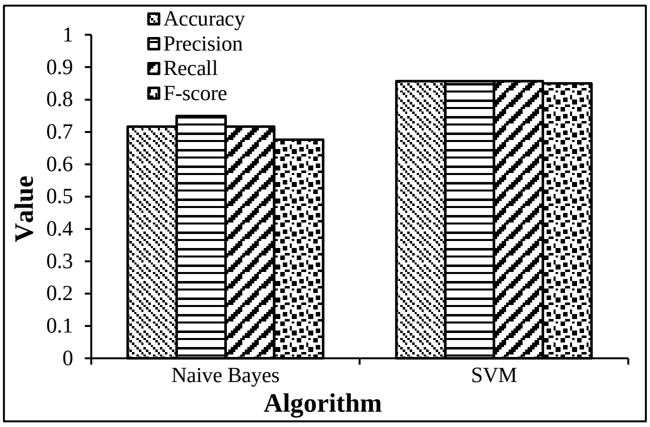

Fig.5. Performance comparison of Naive Bayes and SVM classifier on test dataset

The Fig.5 indicates that the base SVM classifier having 85% accuracy outperformed well when compare to Naive Bayes with 71 % accuracy over all samples of the test data.

4.2.2 Modelling using Ensemble of Classifiers:

In this research work, Bagging ensemble models are built using individual Naive Bayes and SVM model. Bagging-NB and Bagging-SVM are the composite models obtained by aggregating the results of several individual Naive Bayes and SVM classifiers respectively. Adaboost algorithm is used as a part of boosting ensemble technique to boost the performance of base SVM model. The comparative study of accuracy performance of Bagging-NB, Bagging-SVM and Adaboost Ensemble techniques on the test dataset is shown in the Table.3 and Fig.6.

Table.3. Performance of different ensemble techniques on test dataset

Ensemble Classifier	Accuracy	Precision	Recall	F-score
Bagging-NB	0.7261	0.7487	0.7261	0.6823
Bagging-SVM	0.8778	0.8774	0.8778	0.8763
Adaboost-SVM	0.8615	0.8611	0.86	0.8577

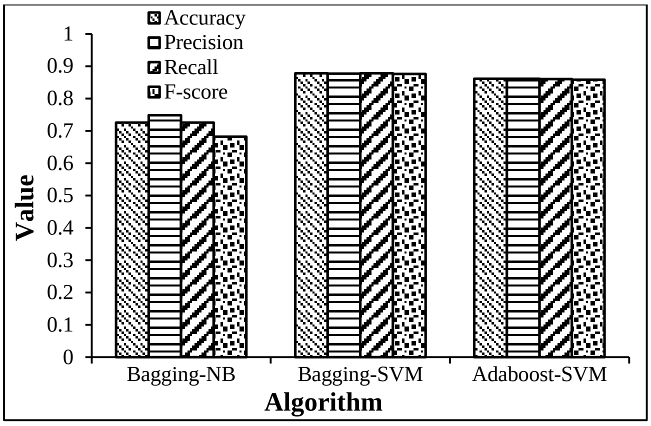

Fig.6. Performance comparison of different ensemble techniques on test data

From the above results shown in Fig.6, we could see that Ensemble of classifier models performs well in comparison with the base classifier models on the test data.

Bagged-NB with 72.61% accuracy achieved good result when compared to individual Naive Bayes classifier having 71% accuracy. Similarly, Bagged-SVM ensemble having 87.78% accuracy outperformed well when compared to the base SVM classifier having 85% accuracy. Adaboost-SVM classifier (86%) also performed well when compared to the single base SVM model.

It is found from all the prediction results that, Bagged- SVM ensemble having 87.78% accuracy performed well when compare to all other chosen model on the test data. So, Bagged-SVM ensemble can be used as predictive model to classify the new unlabelled IT incident ticket.

5. CONCLUSIONS

Manual categorization of IT helpdesk incident tickets may leads to wrong classification and hence causes the assignment of tickets to a wrong domain expert team which in turn causes ticket reassignment, resolution delay and unnecessary use of business resources. To avoid all these difficulties, we proposed a machine learning based ticket classifier system to automatically categorise the IT incidents by mining the unstructured ticket description. Naive Bayes and Support Vector Machines (SVM) are used as base classifiers to build such a predictor model. Ensemble techniques such as Bagging and Boosting are used to build ensemble based incident ticket classifiers. The performance of all the ensemble classifiers is evaluated and compared with the corresponding base classifiers using different performance evaluation metrics. Ensemble classifiers outperformed well in comparison with the individual base classifiers over all samples of test data. In particular, bagging of individual SVM's called Bagged-SVM classifier outperformed well when compared to all other chosen models. A real IT infrastructure historical incident data of a reputed enterprise is considered for this research work. The proposed IT incident classifier system results in proper assignment of tickets to correct support group, effective support resource utilization, improved end user experience, and quicker turnaround time.

REFERENCES

[1] S.D. Galup, R. Dattero, J.J. Quan, and S. Conger, "An Overview of IT Service Management", *Communications of the ACM*, Vol. 52, No. 5, pp. 124-127, 2009.

[2] D. Cannon and D. Wheeldon, "*ITIL Service Operation*", TSO Publisher, 2007.

[3] R. Gupta, K. Hima Prasad and M. Mohania, Mukesh, "Automating ITSM Incident Management Process", *Proceedings of 5th International Conference on Autonomic Computing*, pp. 141-150.2008.

[4] Mucahit Altintas and Cuneyd Tantug, "Machine Learning Based Ticket Classification in Issue Tracking Systems", *Proceedings of International Conference on Artificial Intelligence and Computer Science*, pp. 1-6, 2014.

[5] S. Agarwal, V. Aggarwal, A.R. Akula, G.B. Dasgupta and G. Sridhara, "Automatic Problem Extraction and Analysis from Unstructured Text in IT Tickets", *IBM Journal of Research and Development*, Vol. 61, No. 1, pp. 41-52, 2017.

[6] S. Silva, R. Pereira and R. Ribeiro, "Machine Learning in Incident Categorization Automation", *Proceedings of IEEE 13th Iberian Conference on Information Systems and Technologies*, pp. 1-6, 2008.

[7] S.P. Paramesh and K.S. Shreedhara, "Automated IT Service Desk Systems Using Machine Learning Techniques", *Proceedings of IEEE International Conference on Data Analytics and Learning*, pp. 331-346, 2018.

[8] F. Al-Hawari and H. Barham, "A Machine Learning Based Help Desk System for IT Service Management", *Journal of King Saud University-Computer and Information Sciences*, 2019.

[9] S. Roy, D.P. Muni, J.Y.T. Yan, N. Budhiraja and F. Ceiler, "Clustering and Labeling IT Maintenance Tickets", *Proceedings of International Conference on Service-Oriented Computing*, pp. 829-845, 2016.

[10] C. Cortes and V. Vapnik, "Support-Vector Networks", *Machine Learning*, Vol. 20, No. 3, pp. 273-297, 1995.

[11] T. Joachims, "Text Categorization with Support Vector Machines Learning with Many Relevant Features", *Proceedings of European Conference on Machine Learning*, pp. 137-142, 1998.

[12] M. Ikonomakis, S. Kotsiantis and V. Tampakas, "Text Classification using Machine Learning Techniques", *WSEAS Transactions on Computers*, Vol. 4, No. 8, pp. 966-974, 2005.

[13] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E.D. Trippe, J.B.Gutierrez and K. Kochut, "A Brief survey of Text Mining: Classification, Clustering and Extraction Techniques", *Proceedings of International Conference on Machine Learning*, pp. 1-13, 2017.

[14] M.M. Mironczuk and J. Protasiewicz, "A Recent Overview of the State-of-the-Art Elements of Text Classification", *Expert Systems with Applications*, Vol. 106, pp. 36-54, 2018.

[15] K. Kowsari, K.J. Meimandi, M. Heidarysafa, S. Mendu, L.E. Barnes and D.E. Brown, "Text Classification Algorithms: A Survey", *Proceedings of International Conference on Computation and Language*, pp. 1-7, 2019.

[16] L. Breiman, "Bagging Predictors", *Machine Learning*, Vol. 24, No. 2, pp. 123-140, 1996.

[17] T. Dietterich, "Ensemble Methods in Machine Learning", *Proceedings of International Workshop on Multiple Classifier Systems*, pp. 1-15, 2000.

[18] Y.S. Dong and K.S. Han, "A Comparison of Several Ensemble Methods for Text Categorization", *Proceedings of IEEE International Conference on Services Computing*, pp. 419-422, 2004.

[19] A. Sharma and S. Dey, "A boosted SVM based Ensemble Classifier for Sentiment Analysis of Online Reviews", *ACM SIGAPP Applied Computing Review*, Vol. 13, No. 4, pp. 43-52, 2013.

[20] Yashima Ahuja and Sumit Kumar Yadav, "Multiclass Classification and Support Vector Machine", *Global Journal of Computer Science and Technology Interdisciplinary*, Vol. 12, No. 11, pp. 14-20, 2012.