CHAPTER 2

REVIEW OF LITERATURE

Spam, is a word coined 20 years ago by Cranor and LaMacchia (1998) and is considered as the biggest issue in the digital world and despite the tremendous efforts taken by the researchers all over the world to mitigate this problem, it is still a challenging area and the urgency for a 100% accurate detection system is still remains unchanged. The idea of digital spamming started with spam e-mails and has moved forwards to abuse other techno and social media of communication also. Examples include instant messaging spam, Usenet news-group spam, Web search engine spam, spam in blogs, wiki spam, online classified ads spam, mobile phone messaging spam, Internet forum spam, junk fax transmissions, social spam, spam mobile apps, television advertising and file sharing spam (https://en.wikipedia.org/wiki/ Spamming).

The success and popularity of the social Web in early 2000s, paved way to the introduction of many other new types of spams like Wiki spam (injecting spam links into Wikipedia pages), opinion and review spam and mobile messaging spam (spam messages sent directly to mobile devices)(Ferrara, 2019).The scope of this research work is online review spam identification. Several researchers have proposed methods to detect and remove spam reviews and this chapter presents a literature study conducted to understand the current state of this research area.

2.1. FEATURE SELECTION

Optimal feature vector creation task aiming to achieve maximum performance benefits during disease identification consists of two main steps, namely, feature extraction and Feature Selection(FS) or dimensionality reduction. Both these steps are to be applied in a sequential manner, one after the other, during a classification task. Features are defined as unique attributes, whose discriminative values make an instance and its class in the database.

Increase in the dimensionality of the datasets and the amount of data that is required to produce reliable output has a direct impact on the performance of the Machine

Learning(ML) algorithms. This phenomenon, coined by Bellman (1957) is referred to as the "Curse of Dimensionality". A popular solution used in the ML field is to use a method that searches for a smaller version of the original data that contains only those attributes or features that has maximum information to make the classification successful.

A formal definition of FS algorithm was provided by Guyon and Elisseeff(2003), which identified FS as a task that selects of subset of features that can efficiently describe the input data while at the same reduces the effect of noise and still provide highly accurate classification results. After the removal of irrelevant and redundant data, the output of the FS algorithm is a smaller version of the original dataset that can improve the performance of an ML algorithm.

Creation of an optimal feature vector using FS algorithm is a vital and challenging task in any classification problem (Price *et al.*, 2003) and has great usage with large sized and high dimensional datasets. This fact was proved by the experiments conducted by Somorjai *et al.* (2003). This study concluded that since size of the feature set has direct impact on the computational complexity of a classifier and proper usage of FS algorithms can reduce this complexity during disease identification and classification.

FS algorithms have been used in several applications and an extremely large number of algorithms have been used by researchers and academicians. Several studies have reviewed these algorithms to identify its merits and demerits. Examples of such reviews include Chandrashekar and Sahin (2014), Jain (2017), Venkatesh and Anuradha (2019). Several reviews have been conducted on the FS algorithms that have been applied to specific applications. Examples include Ghojogh *et al.* (2019) who surveyed FS algorithms used for pattern analysis, Hira and Gillies (2015) who surveyed algorithms for selecting optimal features from microarray data. Irrespective of the application, all FS algorithms are considered to have the capability to (Qiang, 2005; Montanes *et al.*, 2003).

(i) select only those features or attributes that have positive impact on the performance of a classifier.

(ii) identify and select only those features or attributes that are relevant and non-redundant.

(iii) retain only those features or attributes that have the highest score according to some pre-defined criterion.

The various methods that are used to select optimal features are reviewed in this chapter and are grouped in three manners and are discussed in the subsequent subsections. They are (i) according to the evaluation function used, (ii) based on the classification labels and (iii) according to the selection strategies used.

2.1.1. Grouping of FS Methods Based on Evaluation Function Used

The feature selection methods can belong to Best Individual Features (BIF), Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) (Xue *et al.*, 2016). Feature selection generally uses an evaluation function that is applied to a single term (or word) (Liu *et al.*, 2015).

Feature selection can be performed using feature reduction or feature selection techniques. Methods like principal component analysis, minimum noise fraction transform, discriminant analysis, decision boundary feature extraction, non-parametric weighted feature extraction, wavelet transform and spectral mixture analysis (Thenkabail, 2016; Awange and Kiema, 2019) belong to the feature reduction category. These techniques aim to reduce dimensions by reducing data redundancy.

The main difference between dimensionality reduction techniques and feature selection technique is that the former uses projection or compression technique to reduce the dimensionality and modifies the original representation of the dataset, while the later merely selects a subset of the original dataset as optimal features without modifying or altering them. Thus, feature selection techniques preserve the original semantics of the variables and thus offer many more advantages during knowledge discovery or interpretability stage of data mining

Optimal features can be selected by scoring individual features (BIF) using measures like information gain (Roobaert *et al.*, 2006) and mutual information (Fang *et al.*, 2015). All these feature scoring methods rank features by their independently determined scores and select top scoring features.

Information Gain (IG) measures the amount of information about the class prediction, if the only information available is the presence of a feature and the corresponding class distribution. Concretely, it measures the expected reduction in entropy. IG chooses those candidate features with more information.

IG concerns how much information each feature can provide. Equations (2.1) – (2.3) are the steps for calculating IG. In Equation (2.1), P_i is the probability of class i, which appears in all N feature and this equation calculates the information of all classes. In Equation (2.2), D_{ji} means that the j^{th} feature contains i kinds of different values. Equation (2.3) calculates IG of the j^{th} feature by finding the difference of Equations (2.1) and (2.2).

$$\text{Entropy}(N) = \sum_{i-1}^{k} P_i \log_k \left(\frac{1}{P_i} \right) = -\sum_{i-1}^{k} P_i \log_k P_i \qquad (2.1)$$

$$\text{Entropy}(D_j) = \sum_{i-1}^{|D_j|} \frac{D_{ji}}{N} \text{ x Entropy}(D_{ji}) \qquad (2.2)$$

$$IG(D) = \text{Entropy}(N) - \text{Entropy}(D) \qquad (2.3)$$

Mutual Information (MI) Criterion is a popular approach to analyze the correlation between features. It measures the contribution of a variable towards reducing uncertainty about the value of another variable. MI was first introduced by Shannon (1948) in the context of digital communications between discrete random variables and was generalized to continuous random variables. MI is considered as an acceptable meter of relevance between two random variables (Cover and Thomas, 1991). MI is a probabilistic method which measures how much information the presence/absence of a term contributes to making the correct classification decision on a class (Guiasu, 1977).

The MI score is particularly useful in feature selection because it maximizes the mutual information between the joint distribution and target variables in datasets with many dimensions. For two features, X and Y, the MI is estimated using Equation (2.4).

$$MI(X, Y) = -\sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) \qquad (2.4)$$

where p(x, y) is the joint probability of X and Y random variables and p(x), p(y) are the probability density functions of variable X and Y respectively. A large value of MI signifies high correlation of two variables. Zero value indicates that two variables are not correlated.

SFS methods (Marcano-Cedeño *et al.*, 2010), select best single features evaluated using a given criterion and then add one feature at a time until the number of selected words reaches the desired K features. SBS (Panthong and Srivihok, 2015), on the other hand, removes a single feature at a time which does not meet a given criterion. The process of removing a feature is repeated until the desired number of features is obtained.

2.1.2. Grouping of FS Methods Based on Classification Labels

Another method of grouping FS algorithms is through the perspective of the availability of labels (target class). In this grouping, the FS algorithms can belong to any of three categories, namely, supervised, unsupervised and semi-supervised methods.

Supervised FS methods are generally used in conjunction with classification tasks (Esmailia *et al.*, 2016).This algorithm exploits the availability of the class labels to select optimal features that have maximum class discriminating ability. In this group, the features are initially generated using training data. Generally, during classification all training data will be used by the learning model to train itself. However, while used with this group of methods, a subset of features are first selected and processed which will be used to train the learning model. This step uses the label information along with the characteristics of the data using BIF methods explained above, to select the final set of features, which are used during training and testing.

Unsupervised FS methods are generally used with clustering algorithm(Feng and Duarte, 2018). The steps involved are similar to supervised FS methods, except that no label information is available during the selection of optimal features and training the learning model. As label information cannot be used to identify feature relevance, this group of methods relies on other alternative criteria for selecting features. Examples of such criteria include (i) a criterion that selects features that can best preserve the manifold structure of the original data (ii) a criterion that seeks cluster indicators through clustering

algorithms and then transform the unsupervised feature selection into a supervised framework. This can be performed in two manners. One way is to seek cluster indicators and simultaneously perform the supervised feature selection within one unified framework. The other way is to first seek cluster indicators, then to perform feature selection to remove or select certain features, and finally to repeat these two steps iteratively until certain criterion is met. Additionally, all the supervised feature selection criterion can also be used with some modification with unsupervised FS methods.

Semi-supervised feature selection is usually used when a small portion of the data is labeled, where both supervised and unsupervised methods cannot perform FS correctly (Sechidis and Brown, 2018).Supervised FS might not be able to select relevant features because the labeled data is insufficient to represent the distribution of the features. Unsupervised feature selection will not use the label information, while label information can give some discriminative information to select relevant features. Semi-supervised feature selection, which takes advantage of both labeled data and unlabeled data, is a better choice to handle partially labeled data. The general framework of semi-supervised feature selection is the same as that of supervised feature selection, except that data is partially labeled. Most of the existing semi-supervised feature selection algorithms rely on the construction of the similarity matrix and select features that best fit the similarity matrix. Both the label information and the similarity measure of the labeled and unlabeled data are used to construct the similarity matrix so that label information can provide discriminative information to select relevant features, while unlabeled data provide complementary information.

2.1.3. Grouping of FS Methods Based on Selection Strategies

In terms of different selection strategies, feature selection can be categorized as filter, wrapper, and embedded models. Filter models investigate indirect performance measures, mostly based on distance and information measures (Liu and Motoda, 1998), which are independent of the classifiers. A filter based algorithm performs FS in two steps. In the first step, features are ranked based on certain criterion. In the second step, features with the highest rankings are chosen. A lot of ranking criteria, which measures

different characteristics of the features, are proposed. They include algorithms which have the ability to

(i) effectively separate samples from different classes by considering between class variance and within class variance,
(ii) analyze the dependence between the feature and the class label
(iii) analyze the correlation between feature-class and feature-feature
(iv) to preserve the manifold structure, the mutual information between the features, and so on.

Filter techniques assess the relevance of features by looking only at the intrinsic properties of the data. In most cases a feature relevance score is calculated and low-scoring features are removed. Afterwards, this subset of features is presented as input to the classification algorithm.

Filter-based techniques can be either univariate or multivariate. Examples of univariate methods include the two sample t-test and Analysis of Variance(ANOVA) (Jafari and Azuaje, 2006). Univariate selection methods have certain restrictions and may lead to less accurate classifiers by not taking into account feature-feature interactions. The application of multivariate filter methods ranges from simple bivariate interactions (Bø and Jonassen, 2002) towards more advanced solutions exploring higher order interactions, such as correlation-based feature selection (CFS) (Wang *et al.*, 2005; Yeoh *et al.*, 2002) and several variants of the Markov blanket filter method (Gevaert *et al.*, 2006; Mamitsuka, 2006; Xing *et al.*, 2001). The Uncorrelated Shrunken Centroid (USC) (Yeung and Bumgarner, 2003) algorithms is another solid multivariate filter procedure, highlighting the advantage of using multivariate methods over univariate procedures.

Advantages of filter techniques are that they easily scale to very high-dimensional datasets, they are computationally simple and fast and they are independent of the classification algorithm. As a result, feature selection needs to be performed only once and then different classifiers can be evaluated. A common disadvantage of filter methods is that they ignore the interaction with the classifier (the search in the feature subset space is separated from the search in the hypothesis space) and that most proposed techniques are

univariate. This means that each feature is considered separately, thereby ignoring feature dependencies, which may lead to worse classification performance when compared to other types of feature selection techniques. In order to overcome the problem of ignoring feature dependencies, a number of multivariate filter techniques were introduced, aiming at the incorporation of feature dependencies to some degree (Blanco *et al.*, 2004). These filter techniques treat the problem of finding a good feature subset independently of the model selection step,

Wrapper methods are computationally feasible only for small feature vectors because they are much more time-consuming as each iteration of the method requires classifier execution and testing (Jirapech-Umpai and Aitken, 2005). These methods embed the model hypothesis search within the feature subset search. In this setup, a search procedure in the space of possible feature subsets is defined and various subsets of features are generated and evaluated. The evaluation of a specific subset of features is obtained by training and testing a specific classification model, rendering this approach tailored to a specific classification algorithm. To search the space of all feature subsets, a search algorithm is then 'wrapped' around the classification model. However, as the space of feature subsets grows exponentially with the number of features, heuristic search methods are used to guide the search for an optimal subset. These search methods can be divided in two classes: deterministic and randomized search algorithms.

Advantages of wrapper approaches include the interaction between feature subset search and model selection and the ability to take into account feature dependencies. A common drawback of these techniques is that they have a higher risk of over-fitting than filter techniques and are very computationally intensive, especially if building the classifier has a high computational cost.

In a third class of feature selection techniques, termed embedded techniques, the search for an optimal subset of features is built into the classifier construction and can be seen as a search in the combined space of feature subsets and hypotheses (Xiong *et al.*, 2001). Just like wrapper approaches, embedded approaches are thus specific to a given learning algorithm. Embedded methods have the advantage that they include the

interaction with the classification model, while at the same time being far less computationally intensive than wrapper methods.

Feature selection using wrapper or embedded methods offers an alternative way to perform a multivariate subset selection, incorporating the classifier's bias into the search and thus offering an opportunity to construct more accurate classifiers. Several of the wrapper methods use population-based, randomized search heuristics (Li *et al.*, 2001; Ooi and Tan, 2003), although also a few examples use sequential search techniques (Inza *et al.*, 2004). An interesting hybrid filter-wrapper approach was introduced by Ruiz *et al.* (2015).

Another characteristic of any wrapper procedure concerns the scoring function used to evaluate each feature subset found. As the 0–1 accuracy measure allows for comparison with previous works, the vast majority of papers used this measure. However, other proposals also advocate the use of methods for the approximation of the area under the ROC(Receiver Operating Characteristic) curve (Ma and Huang, 2005), or the optimization of the LASSO (Least Absolute Shrinkage and Selection Operator) model (Ghosh and Chinnaiyan, 2005). ROC curves certainly provide an interesting evaluation measure, especially suited to the demand for screening different types of errors in many biomedical scenarios.

The embedded capacity of several classifiers to discard input features and thus propose a subset of discriminative genes, has been exploited by several authors. Examples include the use of random forests (a classifier that combines many single decision trees) in an embedded way to calculate the importance of each feature (Diaz-Uriarte and Alvarez de Andre's, 2006; Jiang *et al.*, 2004). Another line of embedded FS techniques uses the weights of each feature in linear classifiers, such as SVMs and logistic regression (Guyon *et al.*, 2002). These weights are used to reflect the relevance of each feature in a multivariate way and thus allow for the removal of features with very small weights.

Partially due to the higher computational complexity of wrapper and to a lesser degree embedded approaches, these techniques have not received as much interest as filter proposals. However, an advisable practice is to pre-reduce the search space using a

univariate filter method and only then apply wrapper or embedded methods, hence fitting the computation time to the available resources.

Apart from filter, wrapper and embedded approaches, several other approaches have also been proposed. Some examples include Rough sets theory (Piramuthu, 2004). Fuzzy logic based method (Lin and Cunningham, 1995), expert system based methods (Penaloza and Welch, 1996), Score-based methods (Wang *et al.*, 2011; Wang and Chu, 2010), Particle Swarm Optimization (PSO) (Mohammad and Omatu, 2011), Mutual Information (Pugalendhi and Sargunadoss, 2011), statistical methods (Hassan and Soliman, 2010). A comparison of various methods used is reported by Lu and Wang (2009).

2.2. FEATURE SELECTION ON SPAM DETECTION

The effect of feature selection on spam review detection performance was studied by Etaiwi *et al.* (2017). This work analyzed the various preprocessing methods that can be applied before classification of ham and spam reviews. The methods considered were bag of words, Part of Speech (POS) tagging, n-gram term frequencies, stemming, stop word removal and punctuation marks filtering. These preprocessing steps affect the overall accuracy of the review spam detection task. This work investigated the effects of preprocessing steps on the accuracy of reviews spam detection. Two different machine learning algorithms, namely, Support Vector Machine (SVM) and Naïve Bayes (NB), were used.

Khurshid, *et al.* (2018) studied the use of ensemble systems for detecting review spams. They used features from both real and semi-real life datasets and extracted 2 linguistic (unique terms and content) and two structural (rating-driven and date-driven) features. Feature selection was done using particle swarm optimization, cuckoo search, greedy stepwise and chi-square algorithms. Experimental results showed chi-square feature selection algorithm as the one that produced best result.

Khurshid *et al.* (2021) proposed review spam detection using ensemble learning in imbalanced datasets. This work developed a framework using different feature selection algorithms along with data balancing techniques.

Sophia and Rajamohana (2020) proved that an essential step during spam review detection is feature selection algorithm and is used to reduce dimensionality of the feature space and to improve the classification accuracy. They used deep learning classifier, Naive Bayes classifier and K nearest neighbour classifier to detect spam reviews. This work also presented a survey on feature selection algorithms for spam review detection using deep learning techniques.

Rajamohana and Umamaheswari (2017) presented a hybrid approach to optimize feature selection process using Improved Binary Particle Swarm Optimization (iBPSO) and Binary Flower Pollination Algorithm (BFPA) during review spam detection. The system was tested using Naive Bayes and K-Nearest Neighbour (KNN) classifiers.

Fazzolari *et al.* (2020) studied the effect of using feature selection to improve the opinion spam detection process. They extracted features like photo count, review count, useful votes, reviewer expertise, average gap, average rating deviation, first review and reviewer activity. The work proposed Cumulative Relative Frequency Distribution algorithm for selecting optimal features. The classifier used was decision trees.

Crawford *et al.* (2016) opined that focus on reducing the feature subset size to a manageable number is very less. They proposed two methods for this purpose. They are filter based feature rankers and word frequency based feature selection. They concluded that the best way to reduce feature subset size upon both the classifier being used and the feature subset size desired. It was also observed that the feature sub-set size had significant influence on which feature selection method is used.

Behjat *et al.* (2013) presented a PSO-based feature selection algorithm for spam detection. The proposed PSO-based feature selection algorithm searches the feature space for the best feature subsets. The evolution of feature selected is determined by a fitness function. The classifier performance and the length of selected feature vector as a classifier input are considered for performance evaluation. Experimental results showed that the PSO-based feature selection algorithm was presented to generate excellent feature selection results with the minimal set of selected features to be caused by a high accuracy of spam classification based on Multi-Layer Perceptron classifier.

Patil and Bagade (2012) presented an online review spam detection system using language model and feature selection. The proposed system initially performed POS tagging and stemming as preprocessing. Then using sentiment lexicon and wordnet along with text mining techniques, the spam content was analyzed. They proposed the use of co-reference based method to select optimal features. The SVM classifier was used during spam detection.

Patel and Thakkar (2014) proposed opinion spam detection system using feature selection algorithm. This algorithm was based on n-gram techniques, which was extended using feature selection algorithm. The Naive Bayes and least square SVM classifiers were used during classification. The feature selection was performed using information gain filter based algorithm.

2.3. MACHINE LEARNING ALGORITHMS

Massive amount of researches have been carried out within the field of detecting spam reviews during the last decade. Most of these researches focused on detecting spam reviews using machine learning algorithms. As the present research work uses machine learning algorithms to design the spam detection system, this section is dedicated to details regarding the various machine learning algorithms available.

Spam detection algorithms collect data from both information-intensive and data rich online reviews (Lim and Maglio, 2018; Lim *et al.*, 2018). This information, when analyzed efficiently, provide better understanding of the functioning of a spammer and help to identify features that can efficiently distinguish between normal and spam reviews. The past few decades have envisaged several machine learning algorithms being the backbone to several applications that analyze huge sized, high dimensional data (Jiang *et al.*, 2017). Currently, machine learning algorithms have been successfully been deployed to solve several challenging problems. In particular, the usage of mathematical and statistical methods, has become more widespread, for analyzing features in order to gain better knowledge of trends used by spammers, which is otherwise hard to identify, analyze and understand (Raouf *et al.*, 2018).

To effectively handle the threat of online spam reviews, leading e-commerce companies like Amazon, Flipcart, Ebay, etc. have implemented machine learning algorithms-based spam detection systems. The reason behind this is that machine learning algorithms have the capability to learn and identify spams by analyzing efficiently the huge number of reviews.

Machine learning, a termed put forth by Samuel (1967), is a field of science that studies algorithms which have the ability to learn implicitly without explicitly being programmer. A mathematical and relational definition of machine learning is (Mitchell, 1997),"It is a computer program that is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E". A machine learning algorithms solves a problem using three main steps, as given below (Russell and Norvig, 2003):

(i) Generation : Generate initial solutions as representation to the problem.
(ii) Evaluation : Evaluate the generated initial solution.
(iii) Decision : If evaluated solution is the one desired, then stop. Else improve solutions and go to step 2.

A machine learning algorithm starts with some initial representation of a solution to a problem, which is commonly generated randomly along with some predefined parameters. Recent decades have shown an incredible growth in the power, flexibility and accessibility of machine learning algorithms, which have proved to provide noteworthy assistance in various range of applications like computer vision, natural language processing, healthcare, machine translation, web usage mining, banking operations and bioinformatics (Ghasemi *et al.*, 2018). In fact, experimental analysis have revealed that the usage of machine learning algorithms, produce results that are far superior to results produced by human experts (Ciregan *et al.*, 2012). Because of their great achievements, several investigators have analyzed the working of machine learning algorithms (Ukkonen and Makela, 2019) and have also proposed various enhancement methods to improve their learning process and accuracy(Sarker *et al.*, 2019). These solutions can be broadly grouped into two categories, namely,

i. Based on the learning algorithm used

ii. Based on their operational similarity

2.3.1. Categorization of Machine Learning Algorithms Based on Learning Algorithm Used

The machine learning algorithms can be divided into eight groups based on the learning style used. They are, supervised learning, unsupervised learning, semi supervised learning, reinforcement learning, multitask learning, association-based learning, ensemble Learning and deep learning. Among these, the most popular are the supervised and unsupervised learning algorithms (Vennapusa and Bhyrapuneni, 2019).

Supervised learning or classification algorithm is a method that learns from a set of historical input data and uses this learned knowledge to classify new input data (Bloch, 2019). Here, the data is a set of features obtained from the image spams. A supervised learning algorithm maps a new input data (test data) to any one of the pre-defined class. To perform this mapping, the supervised learning algorithms uses a function or boundary or rules to the features. The features are then divided into decision regions, where each region is a pre-defined class. All learning algorithms aim to build a classifier that can differentiate the classes (or regions) with accurate boundaries. The effectiveness of a classifier depends on various factors like learning method used, type of output (binary or multiple or multilabel) and whether their nature is statistical or non-statistical. It is always desirable to have a classifier, that fits with the input data well and which can correctly predict or classify the class labels of new features. To achieve these desirable characteristics, it is required that the classifier has good generalization capability.

A conventional supervised learning model (that is, a classifier) has four main components, namely, training dataset, new input data (test data), target label set and classification output. The learning algorithm of a classifier has access to a vector of features and is referred to as feature space. Each row or instance in the feature space has a target label assigned to it. The feature space is normally divided into two groups, namely, training and testing data. Equipped with the training, testing and label sets, the learning algorithm acts as function that identifies one of the pre-defined target class for the testing data.

A supervised learning algorithm, thus accepts a set of known feature set (training data) along with known responses (target labels) and based on the knowledge gained during learning, builds a model that generates target label for new features (testing). During spam/ham classification, the training features are first extracted from the image data, which are then used by the learning algorithm to mimic human experts. This trained classifier then applies the test data to predict the output as either 'ham' or 'spam'. Thus, the classifier used for spam identification has to be designed as a binary classifier. Various methods are used for dividing the feature space into training and testing. Examples include Re-substitution method, Hold-Out method, Bootstrap method and Cross-validation method (Raschka *et al.*, 2018).

Unsupervised learning algorithm or a clustering algorithm is defined as a mathematical study of methods that can identify natural groups that exists within a class of features (Ivancsy and Kovacs, 2006). It is also stated as a set of methods that aggregates features into groups based on similarity/dissimilarity or distance measures . According to Everitt *et al.* (2001), clustering algorithms have the ability to construct sensible and informative classification of an initially unclassified set of data, using the variable values observed on each individual and is the art of partitioning a feature space into a fixed number of subsets, which are called as clusters, in a manner that the features inside a cluster are similar to each other (inter-cluster distance is small) and similarity of features from different clusters is high (intra-cluster distance is large) (Srivastava *et al.*, 2000) (Figure 2.1).

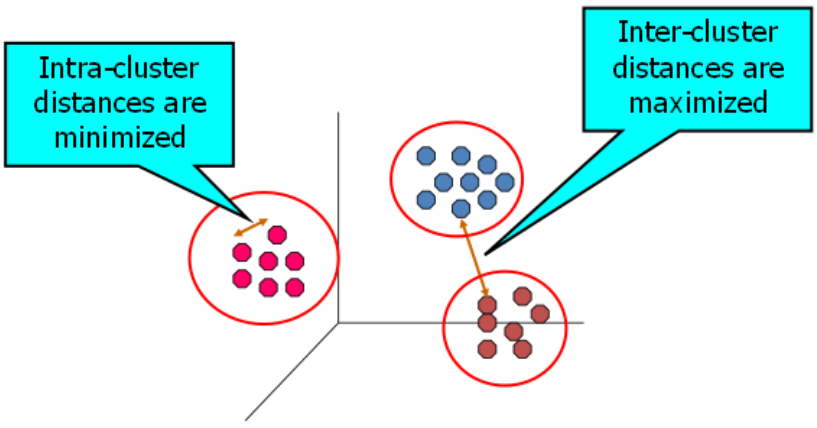

Figure 2.1 : Clustering Process

A clustering algorithm consists of four main steps (Jain and Dubes, 1998), namely, feature vector generation, clustering, evaluation and interpretation. Feature vector generation consists of various steps like extraction, selection, outlier detection and removal and missing value handling. The second step, clustering, uses a distance or similarity function to estimate pattern proximity between two features. The similarity value obtained is to form clusters. Frequently used functions include Euclidean distance, Manhattan distance, Mahalanobis distance, Pearson correlation and cosine measure (Serira *et al.*, 2012; Shirkhorshidi *et al.*, 2015). The similarity function treats the task of clustering as an optimization problem and are well-defined mathematically (Xu and Wunsch, 2005). The clustering algorithms, in general, can be grouped into six broad categories. They are partition clustering, association based clustering, hierarchical clustering, spectral clustering, density-based clustering and grid-based clustering (Rodriguez *et al.*, 2019). All the algorithms depend on the distance measure between two objects and have the common goal to minimize the distance of every object from the center of the cluster to which the object belongs.

After the clusters are formed, the next step is to evaluate their efficiency or quality. Once satisfied clusters are obtained, the next step tries to extract meaningful knowledge from them, which is otherwise hidden. This information can be exploited efficiently by the experts of the relevant field to solve the underlying problems. Additional analysis and experiments can be conducted to attain assurance of the clustering results and its interpretation (Thakare *et al.*, 2016).

Thus, supervised machine learning algorithms perform classification using external assistance and these algorithms learn patterns from training data, the knowledge thus obtained is used on the test data to perform classification. Example problems in this category include classification and regression (Paeglis *et al.*, 2018) and example algorithms include Back Propagation Neural Network, logistic regression, decision tree, Naïve Bayes and Support Vector Machine (https://en.wikipedia.org/wiki/Statistical_ classification).

Consequently, clustering or unsupervised machine learning algorithms learn about data from feature dataset and whenever a new feature data is introduced, the previously

learned feature dataset is used to recognize the class of data. Example problems include clustering, anomaly detection, association and dimensionality reduction. Example algorithms include k-means, DBSCAN and principal component analysis (https://en.wikipedia.org/wiki/Automatic_clustering_algorithms).Both, supervised and unsupervised algorithms, are concerned with the methods that decides on how a search for a good solution can be performed.

Unsupervised learning is normally used to locate patterns in the input data and when supervised learning is used, the precise, correct output which should have been given for any particular training input is known to the system and used by the system to adjust the answer it will give to other training samples (Sathya and Abraham, 2013).

Another category of learning algorithm that combines supervised and unsupervised algorithms is the semi-supervised machine learning algorithm (Burkov, 2019). This set of algorithms can work with datasets that have both labeled and unlabeled data. Examples include generative models, self-training models, low-density separation models, transductive SVM, graph-based models and heuristic approaches (https://en.wikipedia.org/wiki/Semi-supervised_learning).

Andreae (1963) developed a model called STeLLA, which was the first of Reinforcement learning. This model used a trial and error method to find an optimal solution. Improvements on the solution during each trial were done till the desired goal is reached (Salian, 2018). A rewarding mechanism was also installed, which is triggered whenever an action that moves it towards the desired goal is envisaged. The model works on the principle to identify and predict the next step that can earn the final biggest reward. This model, thus, takes advantage of previous knowledge collected through feedback and new tactics that can improve the performance of classification.

Multitask learning models aims to improve the performance of other machine learning algorithms (Hrayr *et al.*, 2019).These models have the ability to remember how a problem was solved and how it reached to that particular conclusion. For this purpose, these models use the knowledge gained through the solutions obtained for other similar problems or task and hence are also termed as inductive transfer models. The working

principal of these models is that when learners share their knowledge, a learning model can produce more accurate and faster results, than when a single learning knowledge is used. Multitask learning models can offer maximum benefits to applications like task grouping and overlapping, exploitation of unrelated tasks, transferring of knowledge and grouping online adaptive learning which has tremendous use in web searching tools and spam filtering. (https://en.wikipedia.org/wiki/Multi-task_learning).

Association rule learning (Issac and More, 2018), or association rule mining algorithms, observe relationships between features to gain knowledge about common patterns that exist in the dataset. These learning methods generate a set of rules that can associate a set of features and are very efficient with huge and high dimensional feature sets that are maintained online by many e-businesses. Frequently used algorithms in this category are the Apriori, FP-Growth and Eclat algorithms (https://en.wikipedia.org\wiki\Association_rule_ learning).

Ensemble systems are models that are composed of multiple weaker baseline models which can be trained independently and whose results are combined using aggregation functions, to obtain the overall output (https://en.wikipedia.org/wiki/Ensemble_learning). The usage of different learning algorithms or different instantiations of the same learning algorithm is termed as an ensemble modeling. It is considered as one of the most powerful to improve the performance of Machine Learning(ML) algorithms in terms of stability and predictive power. An ensemble model allows different and difficult needs of a problem to be handled by algorithms that are best suited to their needs of the application. They have the advantage of providing an extra degree of freedom, thus allowing solutions that would be difficult (if not impossible) to reach with only a single learning algorithm. Moreover, it has been proven that the performance of the ensemble model outperforms the usage of single Machine Learning algorithm (Neeba and Jawahar, 2009; Choudhury *et. al*, 2019). Because of these advantages, ensemble models have been applied to many difficult real-world problems, like, statistics, machine learning and pattern recognition (Brownlee, 2019).

In the past few years, several researchers and academicians have used ensemble model (also called as fusion model) to enhance the performance of ML models (Steinki and Mohammad, 2015). An ensemble model trains different machine learning models on the dataset and then uses them to predict individually. These predictions are then combined, using statistical or algebraic methods, to form the final prediction.

Improving the performance of ensemble models is challenging tasks that has attracted several scholars and have concluded that the task is complex and hence requires lot of effort. An ideal ensemble model can be designed when it is provided with a perfect and accurate set of feature along with a set of classifiers that best suites the application. According to Oza and Tumer, (2008), this task of selecting an appropriate set of learning methods for ensemble construction is a difficult problem. Inspite of several researches having conducted (Park, 2010), the search for an optimal solution is still ongoing. Studies reported have shown that ensemble classification is effective only when the classifiers used in the design of ensemble classifiers exhibit two important criteria, namely, high accuracy and diversity between classifiers (make different error rate).

An ensemble model can be designed in two manners. They are, homogeneous and heterogeneous ensemble models (Elish *et al.*, 2013). Homogeneous models are designed with the same learning methodology trained using different feature sets. These feature sets are created using methods like bagging, boosting, sequential selection and random subspace selection. On the other hand, heterogeneous classifiers are designed with learning algorithms that have different methodology but same feature set. Both types of ensemble systems consist of the following steps, during disease prediction.

(i) Train the machine learning algorithms involved (called as base component)

(ii) Use the learned base component to predict the result

(iii) Combining the individual results

This research work, designs ensemble classification model with the baseline classifier designed as hybrid learning models combining both clustering and classification machine learning algorithms. Studies that used ensembling concept for online review classification is sparse and are generally based on a model built using a classifier

belonging to one type. However, this research work takes advantages of three different techniques, namely, classification, clustering and ensemble technology to improve the identification accuracy of spam reviews.

An advanced set of algorithms, called Deep Learning, that is popular in the recent years can provide more accurate classification results and can construct models that have low complexity (Alom *et al.*, 2019). These algorithms have the advantage of being able to perform classification efficiently with large databases that have complex data belonging to different kinds of data like text, image, video and audio. A list of frequently used deep learning algorithms include Convolutional Neural Network, Recurrent Neural Networks, Long Short-Term Memory Networks, Stacked Auto-Encoders, Deep Boltzmann Machine and Deep Belief Networks (https://en.wikipedia.org/wiki/Deep_learning).

2.3.2. Categorization of Machine Learning Algorithms Based on Their Operational Similarity

The machine learning algorithms can be divided into seven groups based on the operational similarity. They are, regression algorithms, instance-based algorithms, regularization algorithms, decision tree algorithms, Bayesian algorithms, clustering algorithms and artificial neural network-based algorithms.

The first set of algorithm under the second type of categorization is the regression algorithms. These algorithms model the relationship between the features that are iteratively refined with the help of classification errors (Choudhury *et al.*, 2019). Examples of algorithms belonging to this category include Ordinary Least Squares Regression, Linear Regression, Logistic Regression, Stepwise Regression, Multivariate Adaptive Regression Splines and Locally Estimated Scatterplot Smoothing (Li and Wang, 2019).

Instance-based algorithm (Zhang *et al.*, 2019), construct models that gain knowledge using pattern examples obtained from the training dataset and apply the same patterns on the new test data, and thus, deriving the name instance-based. The algorithm first constructs an example database, which is compared when a new feature test data arrives using some similarity or distance measure. For this reason, these algorithms are also termed as memory-based learning algorithms or winner-take-all algorithms

(Daelemans and Bosch, 2005). Thus, the two important tasks of instance-based algorithms are the construction of the representation of the stored instances and selection of appropriate similarity or distance measure to compare instances. Some popular algorithms belonging to this category are K-Nearest Neighbour, Learning Vector Quantization, Self-Organizing Map and Locally Weighted Learning (https://en.wikipedia.org/wiki/Instance-based_learning).

Regularization algorithms are considered as extension to regression algorithms(Gupta, 2019). During the learning process, these algorithms, aim to regularize or shrink or constraint the coefficient estimates to zero. This process makes sure that the model is not complex and avoids the risk of overfitting. Example algorithms belonging to this category are ridge regression, least absolute shrinkage & selection operator, elastic net, least-angle regression (Kayri, 2016).

Decision tree algorithms generate ML models of decisions based on actual values of features or attributes in the data (Shahbaz, 2019). These algorithms construct trees based on training data and use them later during testing or classification. Examples of decision tree algorithms include Classification and Regression Tree, Iterative Dichotomiser 3, C4.5 and C5.0, Chi-squared Automatic Interaction Detection, Decision Stump, M5 and Conditional Decision Trees (Gupta *et al.*, 2017).

Another grouping of ML algorithms based on similarities is the Bayesian algorithms, which is based on Bayes theorem for solving classification and regression problems (Çığşar and Ünal, 2019). Popular algorithms belonging to this category are Naive Bayes, Averaged One-Dependence Estimators, Bayesian Belief Network and Bayesian Network (https://en.wikipedia.org/wiki/Bayesian_classifier).

The next category is the clustering algorithms (mentioned previously as unsupervised ML algorithm), which was described previously, are models that can group similar data without labels. The idea of clustering began with John Snow in 1854, who plotted the diseased reported cases using a special map (Gilbert, 1958). The map reflected closed association between the density of disease cases and a single well located at a central street. This was the first known application of clustering analysis for many

researchers (Tao *et al.*, 2007). Clustering has been increasingly used in several applications like pattern recognition, image processing and information retrieval. Apart from this clustering also has a rich history in other disciplines such as biology, psychiatry, psychology, archaeology, geology, geography, and marketing (Joshi *et al.*, 2011). Frequently used algorithms belonging to this category include K-Means,K-Medians and Expectation-Maximization algorithm.

Artificial neural networks (ANN), is another set of algorithms that has been abundantly used to solve numerous applications (Osisanwo *et al.*, 2019). These algorithms belong to the class of pattern matching and are inspired by the biological neural network of animal brains. ANN algorithms learn to classify/predict by considering examples without using any task-specific rules. Examples include multi-layer perceptron, back propagation neural network, stochastic gradient descent, radial basis function network and hopfield network (https://en.wikipedia.org/wiki/Artificial_ neural_network).

The next category groups algorithms that can combine several classification or clustering algorithms, that is, ensemble systems. This was discussed previously in this chapter.

Finally, a set of algorithms that focus on reducing the complexity of the database are the dimensionality reduction algorithms (Chao *et al.*, 2019). The main goals of these algorithms is to analyze the inherent structure of the data in order to reduce its size and complexity. The removed data are generally have very less value during classification or clustering. Example algorithms of this category include principal component analysis, principal component regression, partial least squares regression, Sammon mapping, multidimensional scaling, projection pursuit, linear discriminant analysis, mixture discriminant analysis, quadratic discriminant analysis and flexible discriminant analysis (https://en.wikipedia.org/wiki/Dimensionality_reduction).

The taxonomy of ML algorithms, presented in this section, is in no way fully complete, as there are abundant number of ML algorithms and many algorithms belong to more than one category (for example: clustering and ensemble). However, the set of algorithms presented in this section are more frequently used in several applications that

include medical imaging and analysis (Fu *et al.*, 2019; Jessica *et al.*, 2019; Kalyani, 2019; Enríquez *et al.*, 2019; Ushmani, 2019;Munkhdalai *et al.*, 2019). The following section presents some studies that use machine learning classifiers for classifying spam and ham emails.

2.4. STUDIES RELATED TO SPAM REVIEW DETECTION

When developing a new review spam detection framework, it is important to understandwhat approaches and techniques have been used in prior studies. This section presents some of the works proposed for spam review detection. From the existing studies, it was deduced that opinion spam detection methods were focused and classified into three main categories i.e. methods based on review centric features, spammer features and spammer group features (Duan *et al.*, 2008; Heydari *et al.*, 2015).

2.4.1. Review-Based Spam Detection

The problem of opinion spam detection was first addressed by Jindal and Liu (2008). They identified three types of reviews for detecting opinion spam, i.e. type 1(fake reviews), type 2(brands reviews), and type 3(non-reviews). They are detected by labeling duplicate spam reviews as positive training sample and other as negative. Their technique achieved good precision on Amazon reviews. However, such type of opinion spam detection has limited scope as the spammers normally change their own reviews to avoid being detected by automated modules.

Ott *et al.* (2011) used supervised learning technique to detect opinion spam. In order to train the learning module, dataset was labeled manually by considering duplicate reviews as positive and rest of the reviews as negative. The technique was supported by various features, i.e., lexical, syntactical and stylistic and used support vector machine as classifier.

Xie *et al.* (2012)studied the singleton review spam detection problem and observed that sudden increase or decrease in rating scores along with increase in number of reviews have got high chances that the rating score is affected by spam reviews. Based on this assumption, a time series pattern discovery mechanism was proposed for spam detection.

Montes-y-Gomez and Rosso (2013) developed a semi supervised learning framework for automatic identification of deceptive and true reviews. This method has the advantage of requiring only a few positive samples and set of unlabeled data.

Prajapati *et al.* (2012) divided the reviews into positive or negative class by doing semantic orientation and using Stanford Natural Language Processing (NLP) parser. After that, they compared the individual review rating with the average rating of that reviews. If the individual rating deviates from average rating, then review was considered as spam, otherwise it was considered as non-spam. Finally, the genuine reviews were summarized by using Chernoff face for the analysis of the precision and accuracy.

Shojaee *et al.* (2013) proposed supervised learning-based classifiers (SVM and Naive Bayes) using lexical features, syntactic features and their combination for spam detection. They used 10-fold cross validation, and it was observed that SVM performs better than the Naive Bayes on all features. However, the major limitation of their work is that they conducted experiments on manually tagged/acquired dataset, instead of real world dataset.

In the same year, Ott *et al.* (2013) used n-gram features in their experiments on SVM classifier on synthetic dataset and achieved an accuracy of 86%. They considered only negative sentiment reviews and compared the performance of results with that of human annotators. They also conducted experiments on Sparse Additive Generative (SAGE) Model based on generative Bayesian approach for detecting fake reviews in three domains, namely: hotel, doctor, and restaurant.

Radulescu *et al.* (2014) proposed a supervised approach to detect opinion spam by three major tasks, namely: (i) feature extraction, (ii) topic extraction and (iii) post comment similarity. After performing feature and topic extraction, similarity between the topic and comment is checked to know, which comment belong to the context of the topic, and at last they have performed classification using three classifiers such as Naive Bayes, SVM, and decision tree. Decision tree outperformed other methods during the experiments conducted on YouTube comments and daily graph news website.

Abu Hammad and El-Halees (2015), used four supervised machine learning algorithms, namely, Naïve Bayes, K-nearest neighbor (KNN), Iterative Dichotomiser 3 (ID3) and Support vector machine (SVM) to detect opinion spam using different features in Arabic review websites. The dataset was compiled manually and labeled as spam or non-spam. The imbalance class distribution problem was tackled by using oversampling method.

Algur and Biradar (2015a) proposed a spam detection technique based on rating and content of the review. Sentiment class and score of the review content is compared with user's given rating. If the deviation between them is greater than two, then it is tagged as spam review. However, inclusion of "review-date-posted" can enhance the performance of the system.

Elli and Wang (2015) applied Multinomial Naive Bayes and SVM classifiers, with bag of words technique. They achieved an accuracy of 80% on the Amazon-based dataset.

Algur and Biradar (2015b) addressed the issue of opinion spam detection (review-based) on different datasets, and used SentiWordNet for assigning polarities to opinion words. They computed spamicity of a given review based on the rating consistency and review content. The training dataset is annotated by the human annotators and achieved an accuracy of 72%. However, more efficient results can be obtained by introducing more features.

Li *et al.* (2017) introduced a novel Sentence Weight Neural Network (SWNN) for assigning weights to user reviews at sentence and document level by proposing a novel method, namely SWNN. They used part of speech tagging and person pronoun features in combination with SWNN in both, cross and mix domains. Improved results are obtained using SWNN in mix-domain. However, more robust results can be obtained, if weight calculation strategy is made in lenient and elastic mode.

Bajaj *et al.* (2017) used personal characteristics for spam detection. This work used the conventional customer ID (name, email ID) feature along with two additional attributes like geographical location and IP address of the device with which the user is

accessing different resources on Internet, to detect spam reviews. In addition, the work also proposed a content analysis method to attack non-reviews using spam dictionary.

Adhav *et al.* (2014) used several features for detecting single review and group review. For this, several features including Untruthful review (includes advertisements),Brand only review(for brand promotions) and Non-review (questions or random texts), were used. The work also discussed the applicability of different techniques like GSRank Method, Factor Graph Model, Behavioral Footprints and Temporal Pattern Discovery, for spam detection.

Crawford *et al.* (2015) used machine learning algorithms for spam review detection, with an emphasis on feature engineering. Moreover, the merits of supervised, unsupervised and semi-supervised learning methods were also analyzed. In feature engineering, review centric features such as bag of words combined with term frequency, parts of speech tag frequencies, syntactic and stylometric features have been discussed for detection of spam reviews.

Jadhav and Gore (2014) used decision tree classifier to identify reviews that were manipulated. Maximum accuracy was achieved when the features are selected using decision tree algorithm. The same authors, in the next year, Jadhav and Gore (2015), used decision tree and SVM to classify the online reviews using features related to sentiment, content, reviewer and product.

Jindal *et al.* (2007) proposed spam detection system based on duplicate finding. To classify, this work designed a logistic regression classifier as a two-class classification problem, non-spam and spam. The work suggested that spam detection should consider both reviews and reviewers.

Rastogi, and Mehrotra(2017) divided opinion spamming into three types based on textual and linguistic, behavioural, and relational features. Moreover, several state-of-the-art machine-learning techniques for opinion spam detection were also discussed.

Saumya and Singh(2020) used an unsupervised learning model combining Long Short-Term Memory (LSTM) networks and Auto encoder (LSTM-Auto encoder) to

distinguish spam reviews from other real reviews. The said model was trained to learn the patterns of real review from the review's textual details without any label. The experimental results showed that this model was able to separate the real and spam review with good accuracy.

Xue *et al.* (2019) proposed content aware trust propagation towards online review spam detection. This work was based on the deviation between the aspect-specific opinions extracted from individual reviews and the aggregated opinions on the corresponding aspects. In particular, the system modeled the influence on the trustworthiness of the user due to his opinion deviations from the majority in the form of a deviation-based penalty and integrated this penalty into a three-layer trust propagation framework to iteratively compute the trust scores for users, reviews, and review targets, respectively. The trust scores were effective indicators of spammers, since they reflect the overall deviation of a user from the aggregated aspect-specific opinions across all targets and all aspects.

Daiv *et al.* (2020) proposed an approach to detect fake reviews using logistic regression based on review centric features. The features used were rating, verified purchase and review length along with review text. Preprocessing methods used were non-alphabetic character removal, stop word removal and stemming.

The aforementioned studies used different features pertaining to the opinion spam detection (review-based) namely: (i) percentage of positive opinion words, (ii) percentage of negative opinion words, (iii) deviation from product rating, (iv) length of the review body, (v) Binary feature indicating if good review was written after first bad review, (vi) size of the review title and (vii) Binary feature indicating, if bad review was written after first good review. However, more features need to be investigated for more efficient classification of opinion spam.

2.4.2. Reviewer-Based Spam Detection

Spammer detection is the process identification of person or group of persons who write fake reviews to promote or demote a target entity. Spammer is a single person or group of persons engaged in writing fake reviews in or against a particular product. Group

spammers can change the opinion on a product and mislead the potential customers, especially in the initial period of product launch, such spammers are highly destructive. Existing work on spammer detection is still in its initial stages.

Mukherjee *et al.* (2011)concentrated on finding group of spammers in three steps: First, candidate spammer group was identified using frequent pattern set by extracting review dataset to isolate product id and review id in each transaction. In second step, spammer indicator values were computed on the basis of eight criteria: Time Window(TW), Group Deviation(GD), Group Content Similarity(GCS), Member Content Similarity(MCS), Early Time Frame(ETF), Ratio of Group Size(RGS), Group Size(GS), Support Count(SC). Finally spammer groups were ranked using SVM rank.

Lim *et al.* (2010) proposed different behavioral models based on unusual review patterns to highlight behavior abnormalities of reviewers. For example, if a reviewer posted all negative reviews on a specific brand competing product, but all positive reviews on a competing product. They concentrated on content pattern and ratings of reviews to formulate four models, i.e., Targeting Product (TP), Targeting Group (TG), General Rating Deviation (GRD), and Early Rating Deviation (ERD). Final spam score was calculated for ranking and evaluating spamming behavior. As this method is dependent of duplications, i.e., same reviewer posts multiple reviews on same product, therefore it is limited for specific type of spamming.

Wang *et al.* (2011) introduced a unique approach used heterogeneous review graph to identify relationship among reviews, reviewers, and stores. The experimental results showed good precision. As this was the first study to use review graph, performance evaluation was performed using human evaluation agreement.

Mukherjee *et al.* (2012) in their extended work, identified two behavior indicators: Group Spam indicator, which is based on above eight parameters and individual spam behavior indicators having following four criteria: Individual Rating Deviation (IRD), Individual Content Similarity (ICS), Individual Early Time Frame (IETF), and Individual Member Coupling (IMC) in a group. This work also used frequent pattern mining to identify spammer group, behavior model was used to show relationship between products,

reviewer-I(individual) and reviewer-G(group)and finally GSRank was used to rank spammer groups. The techniques used were experimented on both supervised and unsupervised learning. The top "n" unexpected rules helped in identifying unusual reviews and reviewers.

Lim *et al.* (2010), Mukherjee *et al.* (2011) and Mukherjee *et al.* (2012) have used behavior model in their work to detect spammers. All the algorithms are capable using of identifying relationship among reviews and reviewers. Algorithms used in these are dependent on duplicates in reviews, where spam reviewer post multiple reviews on same product, except Lim *et al.* (2010), which used relationship graph method to overcome this limitation and can also identify store relationship using heterogeneous graph.. The scoring mechanism is used by all the studies to rank the spammers. The problem of detecting group spammers was addressed by (Mukherjee *et al.* 2011, Mukherjee *et al.* 2012), which are very harmful in changing customer's opinion about a particular product.

Wang *et al.* (2011) identified three features pertaining to review, reviewer and product. Review honesty is the feature depicting how honest the review is. It is based on two factors: reliability of store about which reviews are posted and what are the additional reviews by other reviewers about the same store. Reviewer's trustiness is feature showing how much a user can trust a reviewer; it is based on the accumulative honesty score. Store reliability score is 3rd feature that gives quality of store, if more trustworthy reviewers writes positive reviews, then store is more reliable.

Xue *et al.* (2015) provided an efficient and effective method to identify review spammers by incorporating social relations based on two assumptions that people are more likely to consider reviews from those connected with them as trustworthy, and review spammers are less likely to maintain a large relationship network with normal users. The contributions of this work are two-fold:

- The work elaborated how social relationships can be incorporated into review rating prediction and proposed a trust-based rating prediction model using proximity as trust weight

- Designed a trust-aware detection model based on rating variance which iteratively calculated user-specific overall trustworthiness scores as the indicator for spamicity.

Hazim *et al.* (2018) used statistically based reviewer features for the Extreme Gradient Boost Model and Generalized Boosted Regression Model to evaluate multilingual datasets (i.e., the Malay and English languages). It was observed by the experimental results that the Extreme Gradient Boost Model performed better for the English review dataset and the Generalized Boosted Regression Model performed better for the Malay dataset.

Kumar *et al.* (2018) proposed a hierarchical supervised learning method for spam detection using reviewer features and then characterizing their collective behavior in a unified manner. This method analyzed reviewer's behavioral features and their interactions using multivariate distribution. This work modeled reviewer characteristics and interactions among them as univariate and multivariate distributions and then stacked these distributions using several supervised-learning techniques, such as logistic regression, support vector machine, and k-nearest neighbors yielding robust meta-classifiers.

Zhang *et al.* (2016) recommended a supervised model based on reviewer features to identify spam reviews. This work analyzed the use of non-verbal behavioral features of reviewers and their importance during fake review detection was analyzed. A model pruning based on a sensitivity analysis was used to improved the parsimony of the developed fake review detection model without sacrificing its performance.

The aforementioned studies are based on different spammer-related features, such as (i) review rating, (ii) number of helpful feedback, (iii) number of reviews per day, (iv) reviewer always give same rating, (v) ratio of number of occasion reviewer is the only reviewer, and (vii) ratio of the number of reviews written by reviewer how many time he/she was first and others. However, more robust results can be obtained, if we consider other variations of opinion-spammer features.

2.4.3. Product-Based Spam Detection

Item spam detection is the process of identifying items that are possible targets for spammers.

Wu *et al.* (2010) identified singleton reviews for item spam detection; such reviews are posted by reviewers who write only single review each and post no other review. All the possible spammed hotels in Tripadvisor are analyzed using rating distortion generated by singleton reviews, which helps in isolating true positive from false positive. For this purpose, two scores based on proportion and concentrations of positive singleton reviews are used.

Wu *et al.* (2011) proposed an unsupervised approach to detect spam by taking into account the quality of review, because low quality reviews often result in performance degradation of spam detection. Experiments were performed on the dataset acquired from DangDang website in two domains: Mp3 and book. In order to achieve good quality reviews, the work combined the link analysis and heuristic rules, which performed well in both the domains and its performance was slightly lower than the SVM.

Feng *et al.* (2012) observed that such entities distort their distribution of review scores and leave distributional footprints behind. Experimental results have confirmed the relationship between deceptive reviews and distributional anomaly. This technique can be applied across the domain with minimum cost.

Zhiyuli *et al.* (2015) introduced a novel Sentiment Attribute Matching Stack (SAMS), an unsupervised technique for spam detection, by considering four features, namely: Logistic service, customer service, customer usage and packaging service. Experiments were conducted on a dataset acquired from Taobao.com. An online simulation of shopping system was developed to evaluate the effectiveness of proposed system, which is then compared with the results of manual annotators.

Sun *et al.* (2016) proposed novel convolutional neural network by taking into account the product-related review features for spam detection. This work used bagging technique by combining Product Word Composition Classifier (PWCC), bigram (SVM)

and trigram (SVM) classifier, showing significance performance gain from the three classifiers. The limitation with this work was that they did not consider the review and reviewer related features.

Li *et al.* (2011) used supervised learning and manually labeled reviews crawled from opinions to detect product review spam. This work used the helpfulness scores and comments the users associated with each review. This work was based on the principle that reviews that receive fewer helpful votes from people are more suspicious. Based on this assumption, the proposed method filtered out review data and considered only those reviews which have at least 5 helpfulness votes or comments. They used supervised methods to detect spam reviews.

The aforementioned studies have used different features, such as rank-in-sale and price, which need to be investigated by taking into account hybrid set of features.

2.4.4. Hybrid Spam Detection

The hybrid spam detection techniques combine the aforementioned entities and their features to detect opinion spam efficiently. In following paragraphs, a review of the selected studies in this paradigm is presented

Fei *et al.* (2013) proposed a Novel Kernel Density Estimation (KDE) approach to detect review spam, by taking into consideration the bursty reviews, behavioral features of the reviewer, and the review features. Furthermore, Markov Random filed model with loopy belief propagation algorithm was implemented to track spam reviewer. Experiments were conducted on the dataset used by Jindal and Liu (2008) using unigram features and improved result are obtained.

Noekhah *et al.* (2014) proposed a graph-based unsupervised spam detection technique by assigning priority-based spamicity feature weights. For this purpose, they incorporated feature related to opinion spam (review-based) and opinion spammer (reviewer-based) and obtained an accuracy of 93% on Amazon dataset.

Chen and Chen (2015) proposed a spam detection system by addressing features pertaining to opinion spam detection (review-based) and opinion spammer (reviewer-

based) using dataset obtained from Mobile01. Dataset annotation was performed by crowd sourcing. They used SVM classifier, implemented in python to detect spam reviews.

Rout *et al.* (2017) proposed semi-supervised technique for spam detection by combining different features: sentiment, linguistic, parts of speech, and word count features. Experiments were conducted using the gold standard data set compiled by Ott *et al.* (2013).

Hussain *et al.* (2020) proposed two different spam review detection methods as listed below.

(i) Spam Review Detection using Behavioral Method utilized thirteen different spammer's behavioral features to calculate the review spam score which was then used to identify spammers and spam reviews.

(ii) Spam Review Detection using Linguistic Method worked on the content of the reviews and utilized transformation, feature selection and classification algorithms to identify the spam reviews.

Chaudhary and Shahni (2018) presented an anti-opinion spam detection system for spotting fake reviews using the review sequence of a product. They used features that combined the similarity of personal content of a reviewer, reviewer frequency, repeatability measure and similarity with reviews on a product along with review related features, like review posted date.

2.5. CHAPTER SUMMARY

The current demand of e-commerce industry is to have an accurate and robust spam review detection system that will help them grow their businesses. This chapter presented a literature review of techniques related to the research topic. From the survey, it could be understood that online spam review detection is a multi-disciplinary research field where researchers across many domains are researching. Examples of such fields include text mining, web mining and networking.

From the review, it is understood that the researchers have focused on improving the steps of spam detection, like feature extraction and classification using machine learning algorithms. Researches to improve each of these steps are still active and systems developed using these algorithm are needs to be upgraded to improve the detection process. Travelling in this line, this research work proposes methods that uses multiple feature extraction, feature selection and improved classification algorithm. Methods used to enhance the above steps are introduced along with the research design or methodology in the next chapter, Chapter 3, Methodology.