Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Spam review detection using self-organizing maps and convolutional neural networks

Ashraf Neisari, Luis Rueda, Sherif Saad*

School of Computer Science, University of Windsor, 401 Sunset Ave, Windsor, ON N9B 3P4, Canada

ARTICLE INFO

Article history:
Received 14 October 2020
Revised 16 February 2021
Accepted 16 March 2021
Available online 17 April 2021

Keywords:
Spam review detection
Machine learning
Convolutional neural networks
Self-organizing maps
Word2Vec
GloVe
Fake review detection

ABSTRACT

Online public reviews have significant influenced customers who purchase products or seek services. Fake reviews are posted online to promote or demote targeted products or reputation of the organizations and businesses. Spam review detection has been the focus of many researchers in recent years. As the online services have been growing rapidly, the importance of the issue is ever increasing and needs to be addressed properly. In this regard, there is a variety of approaches that have been introduced to distinguish truthful reviews from the fake ones. The main features engineered in the past studies typically involve two types of linguistic-based and behavioral-based characteristics of the reviews. Unsupervised, supervised and semi-supervised machine learning methods have been widely utilized to perform such a classification. This paper introduces a novel approach to detect fake reviews from the genuine ones using linguistic features. Unsupervised learning via self-organizing maps (SOM) in conjunction with a convolutional neural networks (CNN) are employed to perform classification of the reviews. We transform the reviews into images by arranging semantically-similar words around a pixel of the image or equivalently a SOM grid cell. The resulting review images are consequently fed to the CNN for supervised training and then classification. Comprehensive tests on two gold-standard datasets show the effectiveness of the proposed method on single and multi-domain contexts with accuracy of 88% and 87%, respectively.

1. Introduction

The Internet has enormously grown in size and importance in the past few years, yet it has showed a huge and ever-growing impact on people's daily lives. People spend a significant part of their time surfing the internet to gain information on various topics, communicate with others, and read reviews, articles and news. The Internet also allows people to post reviews about different subjects based on their own knowledge and experience along with others' opinions viewed online. As such, they support or oppose different posts regarding products or services as well. Thus, online reviews play a significant role for both users and providers (Saumya et al., 2018; Singh et al., 2017).

Since anyone can freely post reviews without any limitations, wrongdoers can give undeserving positive or negative opinions to some targeted products, services and businesses. This is done, primarily, to promote or damage the reputation of the target. A person who posts fake reviews is called spammer and his or her posted reviews are called "spam reviews". We also call the reviews posted by a genuine customer, "ham reviews". Spammers are employed, some of them occasionally even by a competitor, to influence reputation, increase or decrease sales of a certain product. Products that show a

* Corresponding author.
E-mail addresses: neisari@uwindsor.ca (A. Neisari), lrueda@uwindsor.ca (L. Rueda), shsaad@uwindsor.ca (S. Saad).
https://doi.org/10.1016/j.cose.2021.102274

higher rate of positive reviews are more appealing for the customers and may lead to increased financial gain for the producer (Saini et al., 2017). On the other hand, products with dominant negative reviews may cause financial loss for the involved companies (Ho-Dac et al., 2013; Zhu and Zhang, 2010). Therefore, truthfulness of posted opinions and reviews need to be examined to avoid misleading the public through deceptive information. In this regard, Ott et al. found in their tests that the average accuracy of three human judges for detecting spam from ham reviews can be estimated to just 57.33% (Ott et al., 2011). In addition, performing such a task manually is a daunting task. Therefore, using automatic detection of opinion spams utilizing state-of-the-art intelligent methods not only does the classification significantly faster, but it also allows to perform it much more accurately than a human expert.

In this paper, we introduce a novel approach to distinguish fake reviews from the truthful ones. This approach uses self-organizing maps (SOM) and convolutional neural networks (CNN) in a special combination. We utilize both supervised and unsupervised learning to extract the hidden linguistic attributes from the reviews. After evaluating the performance of the method on a well-known hotel review dataset (Ott et al., 2013; 2011) and a gold standard Multi-domain dataset (Li et al., 2014), we compare and summarize its performance with other relevant methods. We also examine the effect of using a different prevalent embedding method of, Word2Vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014) of 50, 100, 200 and 300 sizes, on the performance of the developed approach. We performed this by examining these kinds of effects with changes in the sub-components; The effect of choosing the SOM grid size and the neighborhood radius on the SOM unsupervised training. We then used the output of the SOM and feed them to the CNN classifier to distinguish spam reviews from truthful ones.

A related work that combines SOM and CNN was proposed in Fatima and Rueda, 2020, and was used to predict cancer subtypes and stages, yielding very good results on cancer prediction. Unlike this work, Fatima et al. used the SOM to generate a gene similarity network by using gene expressions. In contrast, the method proposed in this paper uses representation learning approaches on text documents and relies on embedding techniques such as Word2Vec and GloVe, which perform a vectorization of high-dimensional data and creates clusters of words that are populated as images that are fed to the CNN. The approach of Fatima and Rueda, 2020, however, creates colored images in which the object or genes are placed distant from each other, not forming any clusters at all.

The contributions of this work can be summarized as follows:

1. Proposed a new method for spam review detection using SOM and CNN.
2. Provided a mechanism to convert any review of variable length into a fixed length numerical array using a SOM.
3. Envisioned and implemented the application of CNN to find the relations of the words in a lower-dimensional space.
4. Developed a Python package for the proposed method, which is available at https://github.com/Ashsari/spam_review_detection.

The rest of the paper is organized as follows. Section 2 discusses the main approaches that have been utilized for spam review detection. The details of the building blocks of the proposed method are discussed in Section 3. In Section 4, we discuss the materials and methods used in this work, including the relevant datasets, data pre-processing and classification methods. We present the results in Section 5, along with a comparative analysis. Finally, we conclude the paper and discuss future work avenues in Section 6.

2. Literature review

Spam review detection was first studied by Jindal and Liu (2007), who used the concept of duplicate and near-duplicate characteristics of the reviews. Since then, there has been a growing interest in this field, as fake reviews have become widespread and impacted various businesses. Due to the challenging nature of the fake review detection task, none of the research done so far has provided a robust solution to the problem, and there is still ongoing research to bridge the gap. Previous works can be categorized based on how different features are engineered and the classification methods utilized.

2.1. *Feature engineering*

Different sources of datasets offer a variety of features available for classification of reviews as spam (fake) or ham (truthful). For instance, the TripAdvisor dataset does not contain characteristics of reviewers, timestamp, rating, and product ID, among others. This shortcoming makes it difficult for the researchers to apply a method devised for a specific dataset to other problems or datasets. Some compiled common features used by previous approaches can be categorized as being related to the review themselves or the characteristics of the reviewers.

2.1.1. *Review content*

This category includes the body of the review related features, reviews content or contextual features. Some studies have used textual content and linguistics features, alone or in conjunction with other features, to assess the veracity of the review. Textual content features include language patterns, used terms, words meanings, and term frequencies, among others. Using textual content alone usually verifies fake reviews with moderate accuracy, e.g., 75% Kohonen and Mäkisara (1989). Spam identifiers utilizing linguistic features can often be fooled by wise spammers trying to write opinions similar to the genuine ones. In addition, textual features are domain-specific, which makes it difficult to create a unified method for cross-domain verification. For instance, the terminology used for describing a restaurant, such as “delicious” or “tasty”, can not be used for car repair shops. Therefore, other features are often employed along with contextual analysis to

boost the detection power and, subsequently, prediction accuracy.

Commonly used methods for feature extraction from relevant corpora include bag-of-words, *n*-gram, term frequency, semantic, sentiment, Part of Speech (POS) Tagging, Linguistic Inquiry and Word Count (LIWC), and Stylometric, among others. Asghar (2016) extracted four types of contextual features, including unigrams, bigrams, trigrams and latent semantic indexing from the body of the reviews and fed them to different machine learning algorithms. They built sixteen models and among all those, logistic regression achieved the highest accuracy of 64%. Shahariar et al., 2019 also selected features from the body of the reviews including *n*-gram, term frequency and word embedding. They reported very good results achieved using deep learning models. Their best reported prediction accuracy is 94.565% using Long Short-Term Memory (LSTM) after splitting the data with a ratio of 70:30.

Saumya and Singh (2018) proposed a method using sentiment of the review in conjunction with other features for the classification. Their proposed method implementing Random Forest (RF) for classification resulted on 91% of F1-score. In a case study, Yilmaz and Durahim (2018) also used the Doc2vec algorithm, which generates document embedding from the textual content of the reviews. This method addresses the issue of the reviews being in a variable length by generating a fixed-length embedding vector for each review to be fed to the classifiers.

2.1.2. *Characteristics of the reviewer*

This subcategory includes information about the reviewer and group-related features. Users' behavior and footprint provide effective features for identifying fake reviews, spammers or spammer groups. These features, when used in combination with other features, have shown better results compared to employing linguistic or behavioral features alone. Compared to the methods that focus on contextual analysis, using the behavioral features reduces computational costs and saves time. The behavioral features used in the past studies include reviewers target products, rating, early posts and ratings, ID, number of posted reviews and rate, review length, polarity mean and distribution, and content similarity, among others.

Mukherjee et al., 2013 introduced a clustering approach by observing the behavioral footprint of the spammers and non-spammers. They hypothesized that these two clusters have different behavioral distributions including content similarity, reviewer maximum number of posted reviews, burstiness, ratio of first reviews, and early time frame, among others. Hussain et al. (2020) used both linguistic and reviewer behavioral-based features in their proposed methods. They extracted thirteen different features based on reviewer behavior including review length, the ratio of first reviews, negative and positive reviews, activity window, review count, and others. They also used these behavioral based features to output a labeled dataset that groups reviews as ham or spam. They used this generated labeled dataset as the input to their second method for identifying spam reviews. In their second proposed method they extracted some linguistic based features and feed them to the same classification models used in the first method. The study showed behavioral based features achieve better results compared to using linguistic based features. Their proposed method using behavioral features yielded 93% accuracy and using linguistic features achieved 88.5% accuracy.

2.1.3. *Metadata features*

This subcategory includes the review itself and product metadata related features. Some studies have utilized clues from metadata and considered reviews characteristics in general rather than just focusing on the content of each reviews. There are datasets that provide additional information needed for this approach. The metadata include price, sales rank, ratio of the positive to negative reviews, and target product ID. These features can help identify anomalous activities. The features of the metadata have been very beneficial, since they help recognize spammers and link them to various domains easily.

Names or IDs in metadata can be used to detect whether a specific product was the spammers' target. Spammers' groups can be spotted by examining the group of products being reviewed by a verified group of users in a short period of time (Rayana Shebuti and Leman, 2015). Additionally, cross-domain spam detection can be achieved if spammers use the same IDs, under the same or different names, to review different targets. The number of feed-backs for the helpfulness of the products and percentage of feedback a review receives are useful in estimating the quality of the reviews (Fei et al., 2013; Jindal and Liu, 2007). Having estimated the ratio of all good quality reviews to bad quality ones, and abnormal ratings of spammers reviews can be detected.

Some studies (Duhan et al., 2017) have also used the location of the reviewer to detect spammers. They extracted the IP address of the reviewers to track the spammers' geo location. If a group of reviews in a specific time window is posted from a specific location, then the reviews are considered spam. The time stamps of the reviews are also a good indication of the spammers' activity (Rayana Shebuti and Leman, 2015). Since some of the fake reviewers post opinions as soon as the product is released or some even before that, Fei et al. used the trait of the burstiness of the reviews in specific unexpected times to classify spam and ham (Fei et al., 2013).

2.2. *Machine learning*

Machine learning methods have been mostly utilized in the past studies for fake review classification. These methods are categorized as supervised (Rayana Shebuti and Leman, 2015), unsupervised (Rayana Shebuti and Leman, 2015), and semi-supervised learning.

2.2.1. *Supervised learning*

Cardoso et al. (2018) presented a comprehensive analysis using supervised methods that utilize textual features. They employed different supervised classification algorithms including multinomial naïve Bayes (MNB), Bernoulli naïve Bayes (BNB), *k*-nearest neighbor (*k*-NN), decision tree (DT), RF, Rocchio, support vector machine (SVM) and MDLText. The performance of existing state-of-art methods vary due to the different engineered features, sentiment of the reviews, domain and the datasets used in the validation phase. The classification performance changed for all the mentioned classifiers when trained and tested differently, whether the reviews had

positive or negative polarity, involved single or various domains, or were tested on different datasets.

2.2.2. *Unsupervised learning*

When labeled datasets are not available, unsupervised classifiers are considered very helpful. Using real-world datasets provided by Amazon, Liu et al. proposed a unified unsupervised framework to detect fake reviews (Liu and Pang, 2018). They used the idea that fake reviewers, usually, have not purchased or used the product they are describing. They are mostly reviewing for financial gain or for changing the reputation of a product. As such, their reviews would show abnormalities and deviation from expected values on many dimensions. The authors proposed the method of Review Deviation based Model RDM2. Their evaluation and analysis resulted in accuracy of 71.18% to 78.62% on different datasets from Amazon. Even though that method showed fair performance, it would need further investigation to consider some deviations that may wrongly affect the results. These deviations are caused by technical terminologies used by some honest experts which regular consumers might not use in their reviews.

2.2.3. *Semi-supervised learning*

In general, supervised learning needs a sizable set of labeled data for training. Providing this amount of labeled data in many applications including spam review detection is very time consuming and costly. However, providing a small set of labeled data is not as costly, and can significantly improve the performance of a classifier as it learns from the data. Such forms of semi-supervised learning methods have also been used for spam review detection.

A two-view method on review and reviewer features was used by Li et al., 2011 to label a huge amount of unlabeled data using a small training set. Using this method there was finally enough labeled fake reviews to train a classifier. These reviews were labeled fake only if both views had regarded them as fake. Fusilier Donato Hernández et al., 2013 used a different approach for fake review detection by implementing "PU" learning. It uses iterative trials of negative classification outcomes. Having removed the positive classified instances, the model was trained on the rest of the unlabeled data and achieved a performance of 83.7% on the *F*-measure.

3. Background

In this work, we use Convolutional Neural Networks and Self-organizing Maps to classify spam and truthful reviews. Here, we provide a brief description for these building blocks of our research.

3.1. *Convolutional neural networks*

As a relatively new twist to the traditional neural networks and inspired by the cognitive neuroscience, the CNN was developed as a type of the deep neural networks. One of the early studies carried out in this area by Hubel and Wiesel's was based on the visual cortex of cat's brain (Wurtz, 2009). They discovered that the visual cortex is largely made up of simple neurons, which respond to small motifs and complex ones that correspond to larger motifs. Later works in this area led to the design of the CNN framework, which works with grid-structured inputs.

Two-dimensional images as the most common grid structured inputs with intense spatial dependencies can therefore be successfully analyzed using CNNs. Grayscale images are two-dimensional grids of pixels intensities, while color images can be represented by three grids of intensities for the three principal colors of red, green and blue, or RGB for short. A typical process for the classification of images can be summarized as follows. **Convolution** is first performed on all regions of the image by applying filters of usually small sizes, 3×3, for instance, to detect small features in the image. Activation of often ReLU-type operates on the convolution output and the result of this process is then called activation map. It is basically revealed in the activation map if the pattern of that filter exists in any region of the image. **Pooling** is then performed to reduce the size of the activation maps resulted from a convolutional layer. This is very important as the size of large images can lead to shear amount of resources needed for processing. This basically determines whether or not simple features exist in larger regions of the picture.

Multiple consecutive layers of convolution and pooling determine the existence of more complex features in the image as it goes through further processing. The size and number of these layers, the hyper-parameters, are selected iteratively based on the performance of the network. **Fully connected** NN layers can then be used to train and learn the relation of those complex features in the image and output the probability for the class of object the image represents.

3.2. *Self-organizing maps*

A self-organizing Map (SOM), also called Kohenen SOM (Kohonen and Mäkisara, 1989), is an unsupervised NN that is mostly used for projecting high-dimensional data points onto a lower-dimensional space, typically, two-dimensional. Unlike other artificial NNs, which employ back propagation through gradient descent and iterative error reduction, the SOM is a competitive learning mechanism. During the learning phase, each node or neuron competes with others to become closer to the input data points. At the end, a map is constructed in such a way that similar input data points are grouped together.

A SOM creates a low-dimensional matrix, a grid map, in which each cell is considered as a neuron. Each neuron has a weight vector of the same size as any of the input data points. This matrix is used to evaluate the distance of any input vector in the dataset from the weights of each cell.

The weights assigned to each cell is randomly initialized at the start of the training phase. Then, iteratively and for each data point, the closest neuron, the one with the smallest distance to the data point, is found and the data point is assigned to it. This neuron is referred to as the Best Matching Unit (BMU). Next, the BMU weights are updated in such a way that the neuron is shifted towards the data point. The weights of BMU neighbouring neurons are also updated during the same iteration and thus they are shifted towards the same direction as the shifted BMU, but with a smaller rate. The amount of changes in the neuron weights decreases as more

iterations are performed during the training, as the grid map of neurons becomes closer and closer to the input data points during every iteration. The process proceeds with all the data points during the training. At the end of the training phase, each data point is assigned to a cell of the SOM grid map. Similar inputs are grouped together around neighboring cells.

4. Proposed method

In this section, we describe the dataset used, the pre-processing task performed on the raw dataset, and the proposed method that uses SOM and CNN for spam review detection. As in most data analysis experiments, the data needs to be pre-processed based on the needs for the specific methods to be used. As such, we cleaned and embedded words from the vocabulary used in all reviews of the corresponding dataset. Word2Vec and GloVe were both used for embedding, and also word frequency and TF-IDF were utilized as the features for supervised training.

The core of our proposed method uses two very powerful types of NNs in conjunction, CNNs and SOMs. While CNNs provide unparalleled capability for image classification, their inputs need to be of fixed size. In our proposed approach we used SOM to convert variable length reviews into fixed size grids or images. The characteristics of the reviews are preserved during the SOM conversion by assigning selected features to the clusters of semantically similar words. To illustrate the concept, two following reviews can be considered, the first a two-word review of "great product.", and the second a two-hundred-word review of any arbitrary content. A word embedding of size 300 for instance then can be used to convert words to numerical values. This makes the input of size 600 for the first review and 60,000 for the second one to a classifier. Using an SOM, we propose, we can convert both reviews into a 20×20 grid or image for instance, suitable for CNNs to classify.

CNNs are comprised of convolution, pooling and fully connected layers of neurons. The convolutional layers are used to detect features in the image while pooling layers make the representations smaller in size. Fully connected layers can then be trained to learn the relations and complex features in the images and estimate the probabilities for the class the images represent.

Unlike CNNs, SOMs are unsupervised NN that are mostly used for projecting high-dimensional data points onto a lower-dimensional space, typically, two-dimensional. During the projection step, similar and related data points are clustered together. In addition, unlike other types of NN, which are trained by backpropagation, the SOM is a competitive learning approach. The details of SOM training, specific to our proposed method, are described in detail in Section 4.2.1.

The proposed method first clusters semantically-related words present in the review corpus to a two-dimensional grid via a SOM training step. The map is then used to represent each review as a unique fixed-size image with different representation layers. Afterwards, the reconstructed review images are fed to a CNN to train and classify reviews as ham or spam. Our approach utilizes linguistic features to extract lexical diversities spread in ham or spam reviews.

4.1. *Pre-processing*

In order to classify reviews as fake or truthful, we first performed some pre-processing steps to make the data ready for the classifier we used. The pre-processing steps we used are described below.

Reviews were first cleaned by removing all the punctuation and special characters, such as "(", ",", "*", and other special characters. Then, all the words were converted to lower case and tokenized as unigram terms. Tokenized words were used to calculate the Term Frequency-Inverse Document Frequency (TF-IDF) of the words and fetch their embedding vectors from the pre-trained dictionary, as described below.

The TF-IDF weight is a statistical measure used to evaluate how important a word is to a document in a collection of texts or corpus. The importance increases proportionally to the number of times a word appears in the document. On the other hand, the importance decreases by the frequency of the word in the corpus. Rare words contribute more weights to the model, whereas highly frequent words would not provide much information gain.

Word embedding is a technique used to represent Natural Language in a fixed-dimension vector of real numbers. This technique automatically captures the essence and relationships among words by employing deep learning. The main idea is that the meaning of a word is given by the words that frequently appear close-by in the text. In our experiments, we have tested two well-known pre-trained word embedding dictionaries, namely Word2Vec and GloVe, and selected the one that performed the best.

Word to Vec (Word2Vec) embedding dictionary includes three million words and phrases out of roughly 100 billion from the Google News dataset (Mikolov et al., 2013). Each word, represented as a 300-dimensional vector, is created by a deep learning model that computes and measures how well a certain word can predict its surrounding words. In this dictionary, some of the stop words such as "a", "and", "of" are excluded, while others such as "the", "also", and "should" are included. The dictionary includes misspellings of words and commonly paired words, i.e., "Soviet_Union" and "New_York".

Global Vectors (GloVe) is a word representation dictionary developed at Stanford University with 400,000-word vocabulary trained on a six billion token corpus from Wikipedia 2014 + Gigaword 5 (Pennington et al., 2014). GloVe provides pre-trained word vectors of 50, 100, 200 and 300 dimensions. Unlike Word2Vec, which only considers local contexts, GloVe captures the meanings in the vector space and takes global count statistics into account.

4.2. *Detection of spam reviews*

We have designed an approach that is used to distinguish between spam and ham reviews. The method is based on SOM and CNN, and takes advantage of the strengths of each method: the SOM allows for representation learning and dimensionality reduction, while the CNN captures the spatial, hidden relations embedded in the features, which are useful for image classification. Since it is almost impossible to figure out the relationships in the input vectors in such a high-dimensional space, the SOM helps represent the data onto

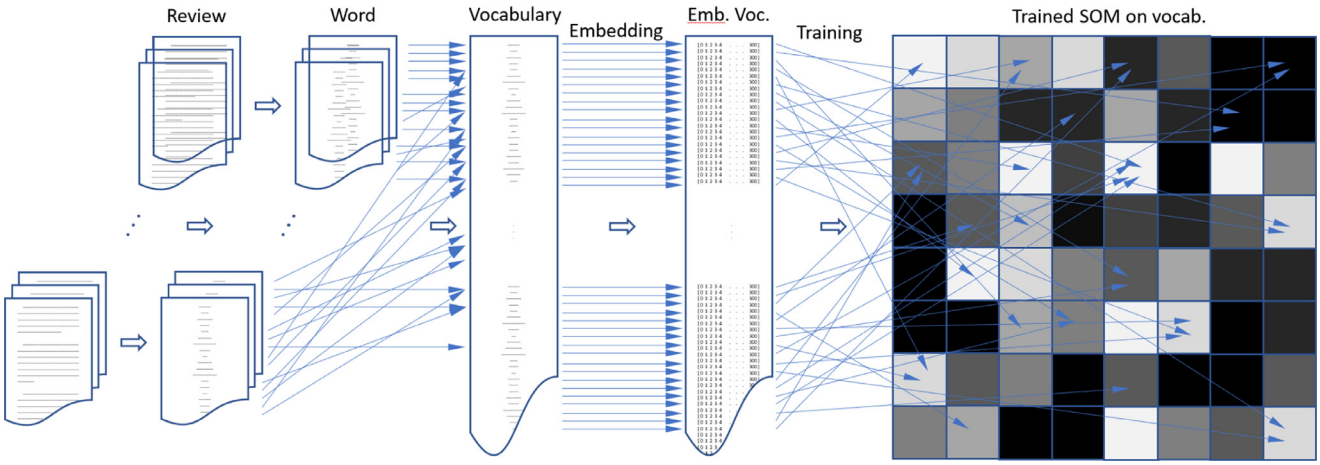

Fig. 1 – Step 1: Construction of the SOM grid map.

a lower dimension, typically, a two-dimensional map, while preserving the relationships of the input data in the lower-dimensional space. The data can then be distinguishable by the CNN provided that an image representation is used. The main steps carried out by the proposed method are described below.

4.2.1. *Step 1: Constructing the SOM*

The procedure for this step is depicted in Fig. 1, formulated for this work based on a SOM clustering application (Kohonen and Mäkisara, 1989; Vesanto and Alhoniemi, 2000). Each review is pre-processed and the words used are tabulated in separate lists. Then, the unique words from all reviews' lists are tabulated in a vocabulary list, $V_{i\ i=1\to m}$, where m is the number of the unique words in the vocabulary. The embedding matrix, $E_{ij\ i=1\to m,\ j=1\to q}$, is subsequently constructed, where q is the size of the word embedding. Each row of the matrix represents the corresponding word embedding vector. The vector is fetched from a pre-trained embedding dictionary of fixed size q such as Word2Vec or GloVe. A SOM, namely an arbitrarily sized, n, grid map of cells, $C_{ij\ i=1\to n,\ j=1\to n}$, is then used to cluster all words in the vocabulary using an unsupervised learning algorithm. Clustering happens as each word, represented by its equivalent embedding vector, is assigned to a cell of the SOM grid during the training process.

Each cell of the SOM grid, C_{ij}, contains a randomly initialized weight vector, $W_{ij\ i=1\to n,\ j=1\to n}$, of the same size as the word embedding, , where

$i = 1 \to n,\ \ j = 1 \to n.$

In the training phase, each word in the vocabulary, V_k, is examined to determine which cell, C_{ij}, it is closest to using its word embedding vector based on Euclidean distance:

$$V_k \in C_{ij} \quad | \quad d = Min(d_{ij}) \quad , \tag{1}$$

$$d_{ij} = \left\| V_k - W_{ij} \right\| = \sqrt{\sum_{l=1}^{q} (V_{kl} - W_{ij,l})^2} \quad , \tag{2}$$

where $i = 1 \to n,\ \ j = 1 \to n.$

The word is then assigned to that SOM cell, also referred to as the *best matching unit* (BMU). The weights of the BMU and its close-by cells are then updated and, as a result, they are pushed towards the corresponding word embedding vector as follows:

$$W(t+1) = W(t) + \Theta(t)L(t)[V(t) - W(t)] \, , \tag{3}$$

where $t = 1, 2, \ldots, s$ is the iteration number out of total s iterations. Θ and L are the neighborhood function and the learning rate respectively, which are both decaying values as the number of iterations increases, defined as:

$$L(t) = L_o e^{-t/\lambda} \, , \tag{4}$$

$$\Theta(t) = e^{-d^2/2\sigma^2(t)} \, , \tag{5}$$

$$\sigma(t) = \sigma_o e^{-t/\lambda} \, , \tag{6}$$

where d is the Euclidean distance between the close-by cell and the BMU, λ is a configurable decay constant, and the initial values of L_o and σ_o are configurable parameters for the training phase, all referred to as hyperparameters. These influence how intensely the weights of the BMU and neighboring cells are pushed towards the word embedding vector.

The last equation above is particularly interesting as it shows not only neighboring cell weights are updated much less along the training, but also that the number of cells considered as neighbors is reduced as the neighborhood radius decreases.

At the end of the training, each word of the vocabulary is assigned to a SOM cell. That does not necessarily mean that all SOM cells contain words assigned to; there maybe some cells without any assigned word. That likelihood increases with the size of the SOM grid selected, n.

The number of words assigned to a single SOM cell is a measure of how populated a cluster of words is. At the same time, that alone may not show how apart the words are from each other.

4.2.2. *Step 2: Converting reviews to images using the SOMs*

The procedure for this step is shown in Fig. 2. In this step, each review is converted into a single or multiple layered image or

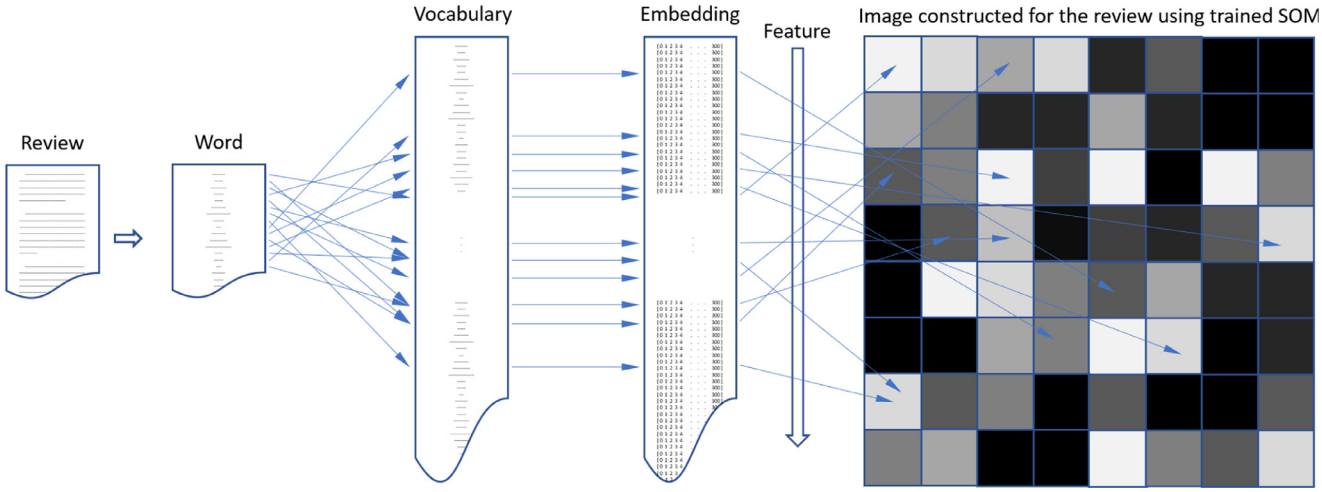

Fig. 2 – Step 2: Conversion of each review to an image.

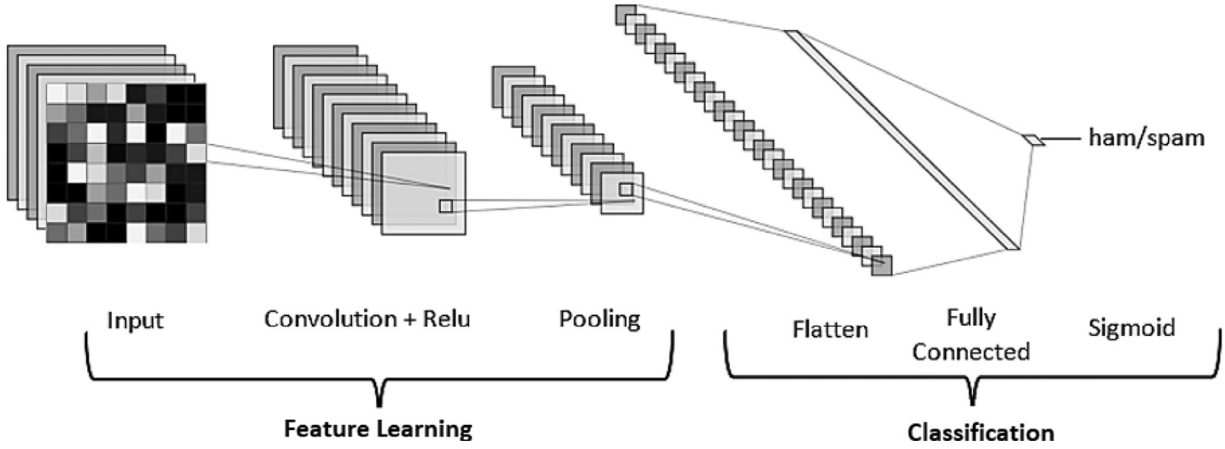

Fig. 3 – Step 3: CNN training on review images and classification.

matrix. The image is of the same size as the SOM grid map constructed in Step 1. The intensity of the pixel, initially zero, is added to the value of the word feature considered for that layer of the image. Each layer of the image corresponds to a different feature of the reviews.

The density of each pixel for a layer is the sum of the quantized measure of the feature chosen for all the words associated with that pixel. We used the number of distinct words in a review as the feature selected for the first layer of the image. TF-IDF of the unique words used in the review was used as the feature for the second layer of the image.

4.2.3. *Step 3: Training the CNN*

The procedure followed in this step is depicted in Fig. 3. The images constructed in Step 2, which represent the reviews, are fed to a CNN for the supervised training and then classification. The CNN is a well-known supervised deep learning algorithm that detects and learns the hidden attributes of the training data, especially images. Thus, the CNN is an excellent choice to classify the generated review images using their corresponding labels provided in the dataset.

The CNN architecture used in this work is a stack of convolution, pooling, flattening, fully connected and a dropout layer. In this architecture, we apply 32 filters of size 3x3 in the convolution layer, followed by an activation function of ReLU. Next, we apply a sub-sampling (Max-pooling) layer of 2×2 to make the feature maps smaller and more manageable. The flattening layer is applied to reshape the structure of the representations obtained from the pooling layer. In the first fully connected layer, we assign 100 hidden neuron to reach the classification decision. Consequently, we apply dropout to prevent the network from co-adaptation of the features. At the end, we apply a fully connected output layer that uses the Sigmoid function to output the final probabilities for each class.

4.2.4. *Step 4: Classification of new reviews*

Once both the SOM and the CNN used in our method have been constructed and trained, a new review can be classified using the procedure depicted in Fig. 4. This classification is also used for evaluating the classifier through training and testing accuracy.

First, the new review is converted into a (multilayer) image in the same way as described in Step 2 using the trained

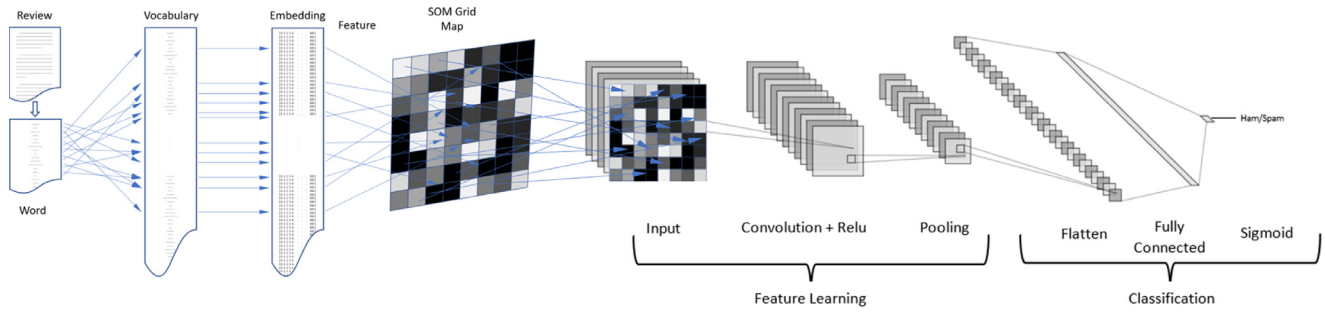

Fig. 4 – Step 4: Classifying new reviews.

SOM. The resulting image is classified using the trained CNN obtained in Step 3.

5. Results and discussion

A number of experiments have been conducted to evaluate the performance of the proposed method. We performed our experiments using Python language in Google Colab-Pro environment and set the runtime to use GPUs. For SOM deployment, we used the Minisom version 1.1.2 library (Vettigli, 2020) to build the cell grid map and to cluster words utilizing the vocabulary embedding matrix. For CNN deployment, we used Tensorflow, a high-level API of "tensorflow.keras" to build the CNN and train it. Tensorflow version 2.2.0 was used in our work, which was the Google Colab default at the time we conducted our experiments.The CNN architecture used in our experiments contains a convolutional layer with 32 filters of size 3×3, followed by a 2×2 Max-pooling layer. Furthermore, we applied a flattening layer, a fully connected layer containing 100 neurons, a dropout of 0.5 and finally a two-neuron fully connected layer to calculate the probability of each class for ham and spam reviews.

Through the set of experiments for performance evaluation and comparison, we tested our method with two different embedding techniques. We used pre-trained word embedding dictionaries of both Word2Vec and GloVe, in four different dimensions of 50, 100, 200 and 300, to fetch the corresponding word vectors for our vocabulary. To find the best SOM to use, we employed a grid search on different cells map sizes and the neighbourhood radii. SOM map sizes of 20, 30, ..., 100 squared and neighborhood radii of 1, 2, 3, 4, 5, 7 and 9 were tested to observe the effects in performance.

Here, we provide the results of the SOM training in Fig. 5 for illustration. This is for the case of SOM grid size of 20×20, neighboring function starting at 5 and GloVe-300 embedding used. The figure shows unsupervised learning and clustering of all words in the vocabulary, and hence it does not have anything to do with the truthfulness of a review, or whether or not the review is spam. The purpose of this illustration is to show that the CNN can learn patterns of association of a word with probability to occur in spam or ham reviews, and use it to classify a new review as spam or ham. Green cells in the figure contain words appearing in ham reviews only, while red cells contain words appearing in spam reviews only. The black cells do not contain any words, while the other cells which are colored with the mixture of red and green contain words from both ham and spam reviews. Some of the clusters are highlighted with blue and purple for the words used in ham and spam reviews respectively.

We have also selected two review images constructed based on the SOM, one from spam reviews and another from the truthful ones, shown in Fig. 6. The different sizes of green and red circles inside the grid cells represent the number of words from one review assigned to each cell.

We deployed two CNN architectures and selected the one resulting the best performance as reported in Section 3.3.3. The Keras Tuner framework was also used to test and find the best hyperparameters for our model. The CNN architecture chosen still provided better results. Furthermore, we implemented the early stopping mechanism to avoid over-fitting and under-fitting along the training data. This mechanism stops the training at the point in which the performance of the validation set starts to degrade. We used the following metrics: Accuracy, Precision, Recall and F1-Score to evaluate the performance of the method for different combinations (Fei et al., 2013). Also, the CNN was trained by splitting the data into 80% for training and 20% for testing. To make sure the results obtained were not biased toward a special case, we repeated the same tests using K-fold cross validation.

We summarize the results after applying the proposed approach on the Single-domain dataset, described in Section 5.2, and the results on the Multi-domain dataset, described in Section 5.3. The comparative results between our proposed method and the other other approaches are discussed in Section 5.4.

5.1. *Datasets*

We tested our method on benchmark datasets, one proposed by Ott et al. (2013, 2011) and the other proposed by Li et al. (2014). The dataset proposed by Ott et al., which we call Single-domain, is a repository that contains 1,600 hotel reviews. This dataset contains 800 truthful (ham) reviews of both positive and negative polarity, 400 each, and 800 spam reviews, 400 belonging to each polarity as well. The truthful reviews were collected from the TripAdvisor website, and correspond to the 20 most popular hotels in Chicago. Ott et al. recruited a group of people from Amazon Mechanical Turck (AMT) to write fake reviews for the same hotels. The lexical diversity of the ham and spam reviews for this dataset are summarized in Table 1. It can be deducted from the table that

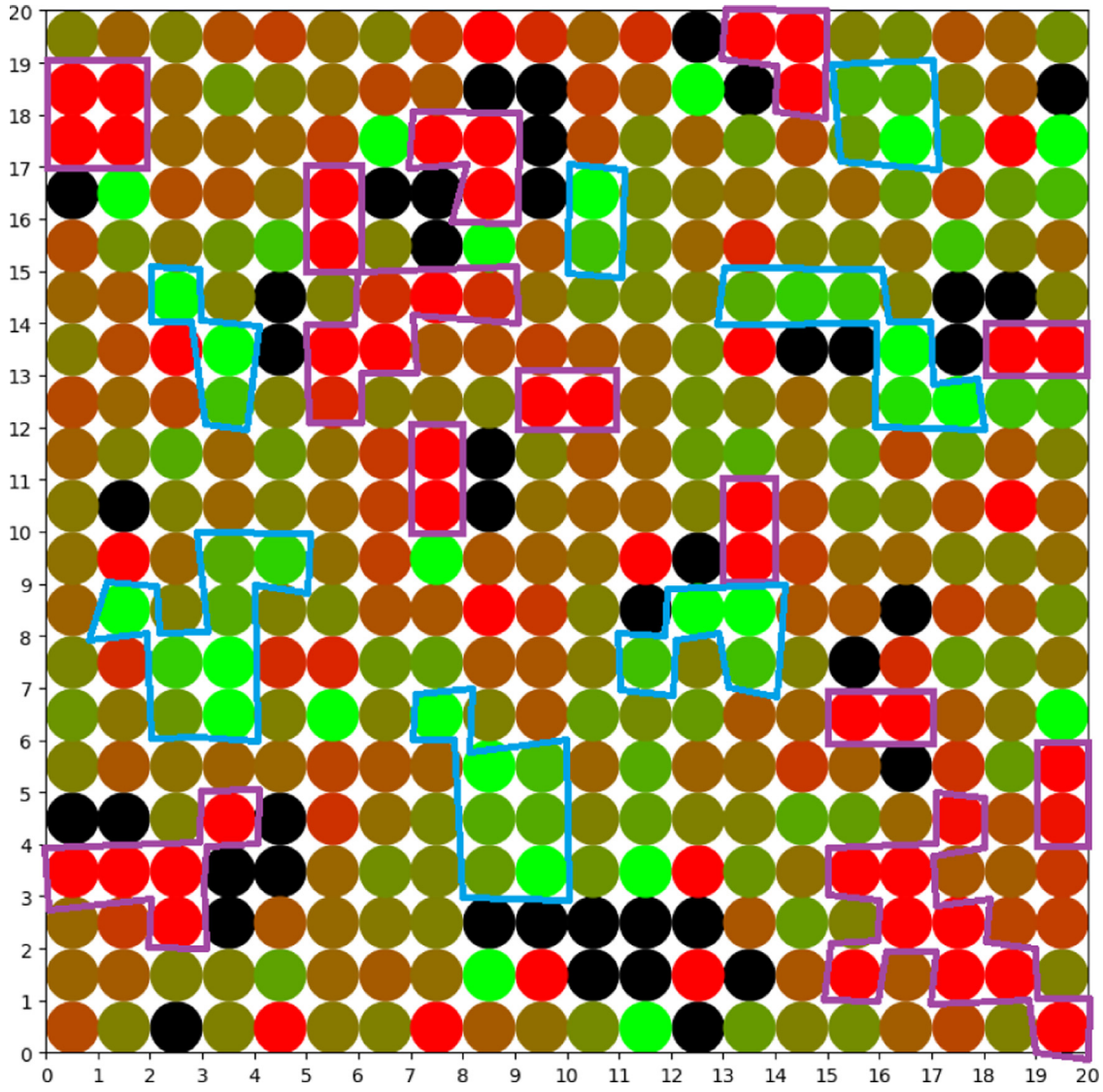

Fig. 5 – Distribution of the vocabulary words arranged into a 20 x 20 SOM grid.

the unique words used only in each ham or spam reviews can also be used as features learned by the machines. CNNs for instance can be used to extract these unique features for classification, yielding superior results.

The second gold standard dataset we used is comprised of reviews from three domains of doctors, hotels and restaurants; we call it Multi-domain. Li et al. (2014). collected these data from three different groups of reviewers. The first group of reviews was written by real customers, the second group was written by the AMTs, and the third group was written by doctors, hotels and restaurant employees who are domain experts. Some statistics for this dataset are summarized in Table 2. The table shows a summary of the unique word counts extracted from a corpus of 2,840 cleaned reviews. We selected this dataset to frame and test the effectiveness of our proposed method on a multi-domain scenario, and to compare the performance of the method with that of a single-domain dataset.

5.2. *Single-domain dataset results*

Fig. 7 shows the performance of our method using GloVe-300 embedding. The effects of the size of the SOM grid and neighborhood radii on the method performance, validation accuracy in this case, are shown in the figure. The results for similar experiments using 10-fold cross validation are summarized in Fig. 8 as well. In general terms, we deduct that the higher the size of the SOM grid, the better our method per-

Table 1 – Lexical diversity in the single-domain dataset.

	Ham reviews	Spam reviews	All reviews	Ham only	Spam only	Common
Unique words count	6355	7214	9604	2390	3249	3965

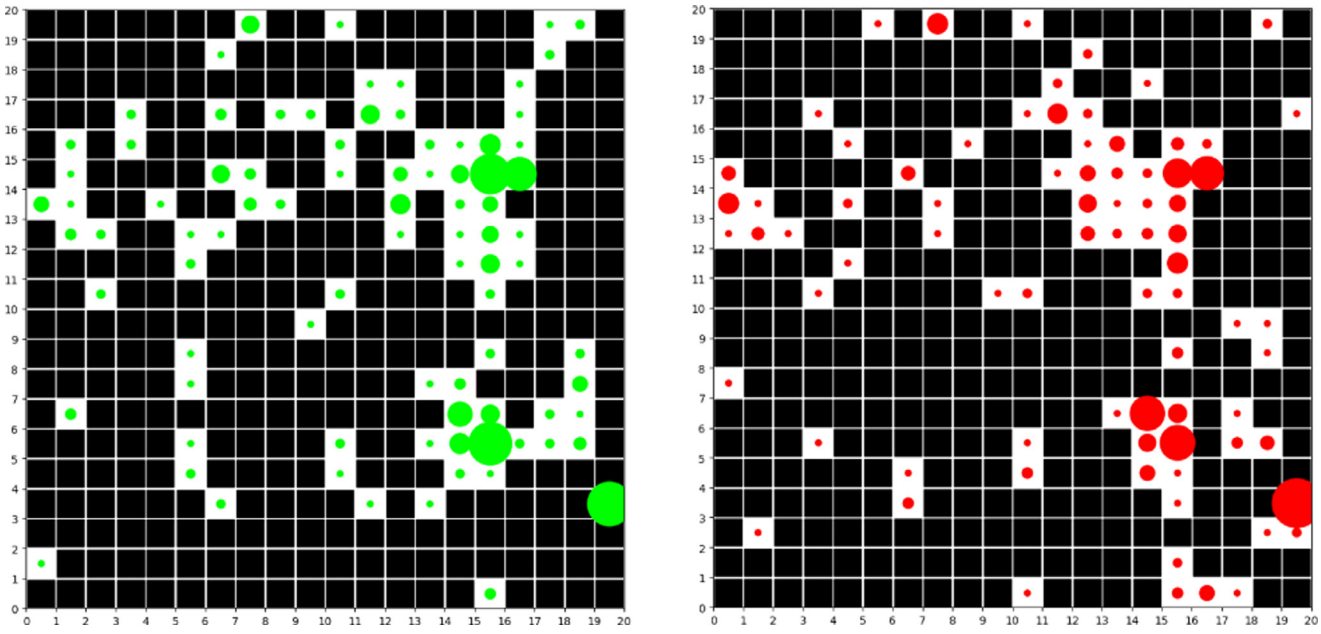

Fig. 6 – Distribution of the words in ham (left) and a spam (right) reviews into a 20 x 20 SOM grid.

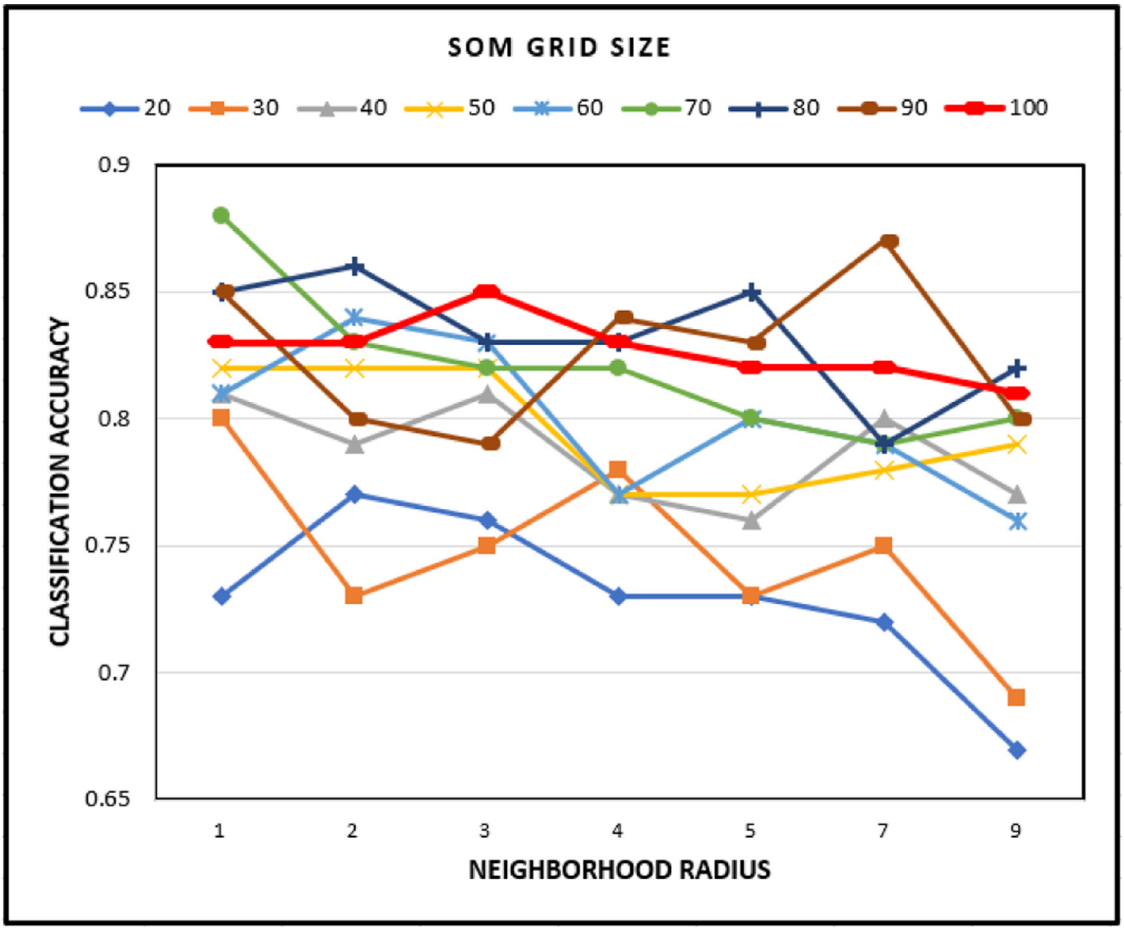

Fig. 7 – The effect of SOM grid size and neighborhood radius on the performance using GloVe-300.

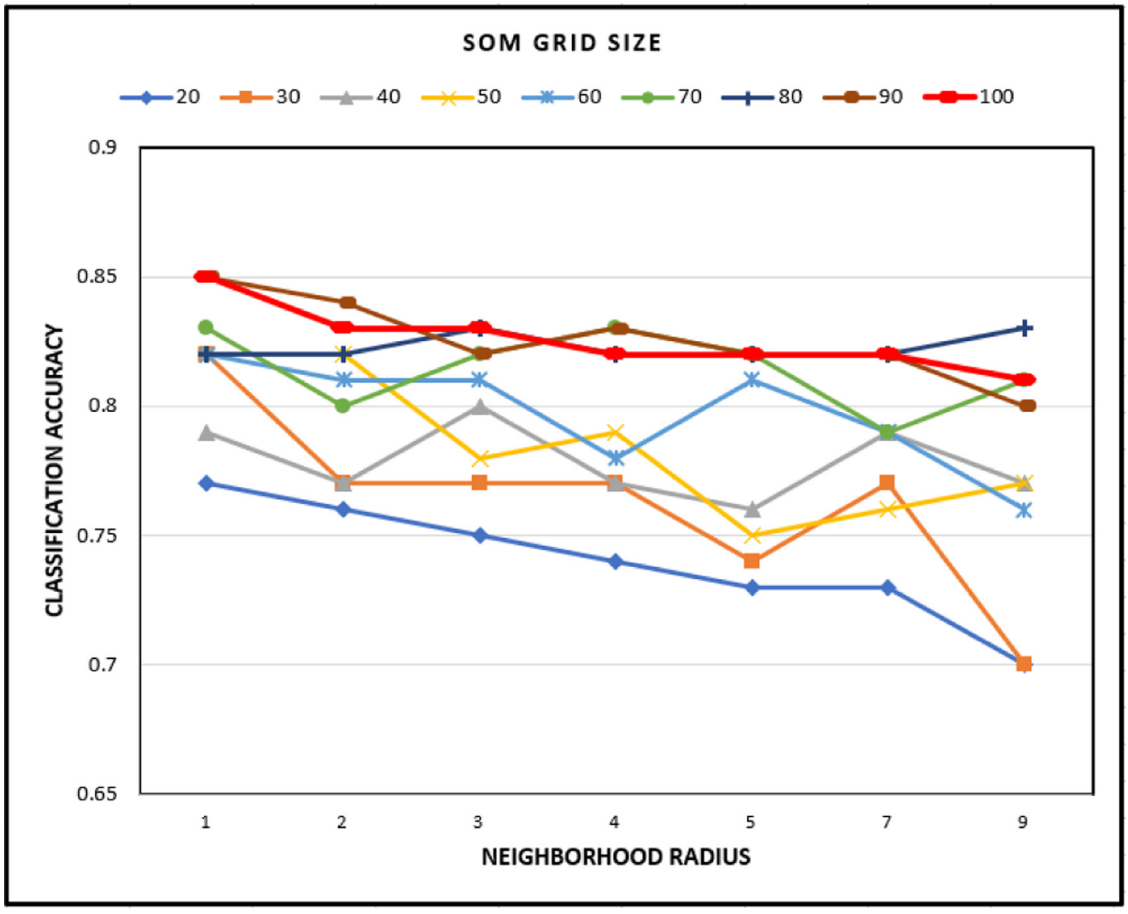

Fig. 8 – Effect of SOM grid size and neighborhood radius on the performance using GloVe-300, 10-fold cross validated.

forms. Similarly, smaller neighborhood values result in better performance.

Observing the results, higher accuracy for a larger SOM grid can be attributed to the effect on the resolution of the map. When the size of the grid map is smaller, the SOM forcefully places less semantically closer words into the same cluster of cells. In contrast, when the grid map is larger, the SOM finds more room to cluster words that are semantically closer, or equivalently similar on a higher-dimensional space. Since we used these grid maps to create the review images and feed them to the CNN for the classification, the more precise and detailed these images are constructed, the better CNN learns their hidden attributes, which subsequently produces better classification as ham or spam.

Similarly, the neighborhood sigma values are related to the dimension of the SOM grid maps. When the dimension of the grid map is small, similar words are densely distributed among the cells. Thus, the smaller the value of sigma is, the more accurate of the model is, since the SOM clusters words in a more compact manner. In contrast, when the SOM grid map is larger, similar words are more widely distributed among neighboring cells. Therefore, not only smaller values of sigma result in higher accuracy, but also higher values of sigma lead to good results.

The results for the same experiments using Word2Vec embedding are summarized in Figs. 9 and 10. While the performance measure is not as high as when using GloVe-300, we observe similar behavior here.

While we have tested different embedding methods of Word2Vec, GloVe-50, GloVe-100, GloVe-200 and GloVe-300, we provide the results for the same 300-dimensional vectors created by Word2Vec and GloVe. In Fig. 11, we show the effect of the embedding method on the performance of the classification, evaluated via different metrics. SOM size of 70×70 and neighborhood function equal to 1 were selected and the highest accuracy reached in our experiments reported here. GloVe-300 yields the best results in most of the cases. In contrast, choosing the second best can marginally depend on the performance metric chosen.

Table 2 – Lexical diversity in the multi-domain dataset.

	Ham reviews	Spam reviews	All reviews	Ham only	Spam only	Common
Unique words count	8986	9124	12,728	3604	3742	5382

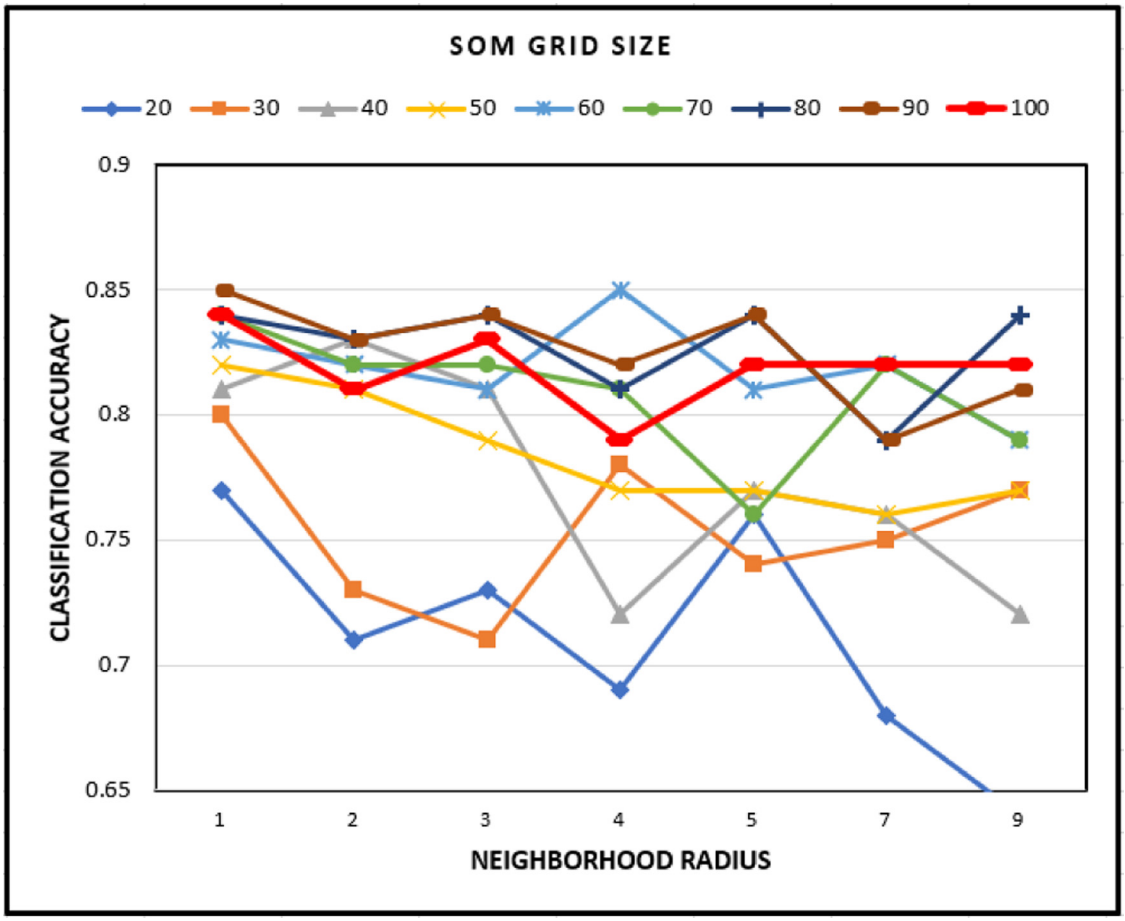

Fig. 9 – Effect of the SOM grid size and neighborhood radius on the performance using Word2Vec.

5.3. *Multi-domain dataset results*

Similar to the Single-domain dataset, Fig. 12 shows the performance of our method using GloVe-300 embedding on the Multi-domain dataset. The effect of the SOM grid size and neighborhood radii on the method test accuracy are shown in the figure. For the Multi-domain dataset, we observe the same trend as that of the Single-domain dataset.

We have also explored the performance of the proposed method on the Multi-domain dataset as a whole and separated as individual domains. The metrics are shown in Table 3, corroborating the effectiveness of the proposed method on a multi-domain dataset. The best performance we observe is the one obtained through grid search, which uses a map size of 70 and a sigma radius of 2. The review images were randomly split into training and testing sets with a ratio of 80:20. Using these best SOM parameters, we further applied grid search to tune the other hyperprameters. For the Multi-domain dataset, we utilized the Keras functional API in order to obtain multiple inputs; the first inputs to a CNN layer and the second inputs to a dense Neural Network (NN). Further, the output of the both networks were combined using a dense layer. This resulted in a boosted accuracy of 0.82%; the results listed the last row of Table 3. Five-fold cross validation was also applied and the results are shown in Table 4.

5.4. *Comparison with other methods*

This section presents a comparative analysis of the proposed SOM-CNN method with other methods used with different types of NN algorithms. The other NN methods used in the literature include CNN, RNN and GRNN, among others. We present the comparisons for the Multi-domain and Single-domain datasets here.

Table 3 – Performance metrics for SOM-CNN on the multi-domain dataset.

	Accuracy	Precision	Recall	F1-score
Hotel	0.86	0.85	0.86	0.85
Doctor	0.94	0.93	0.93	0.93
Restaurant	0.88	0.89	0.87	0.88
Multi-domain	0.82	0.82	0.81	0.82

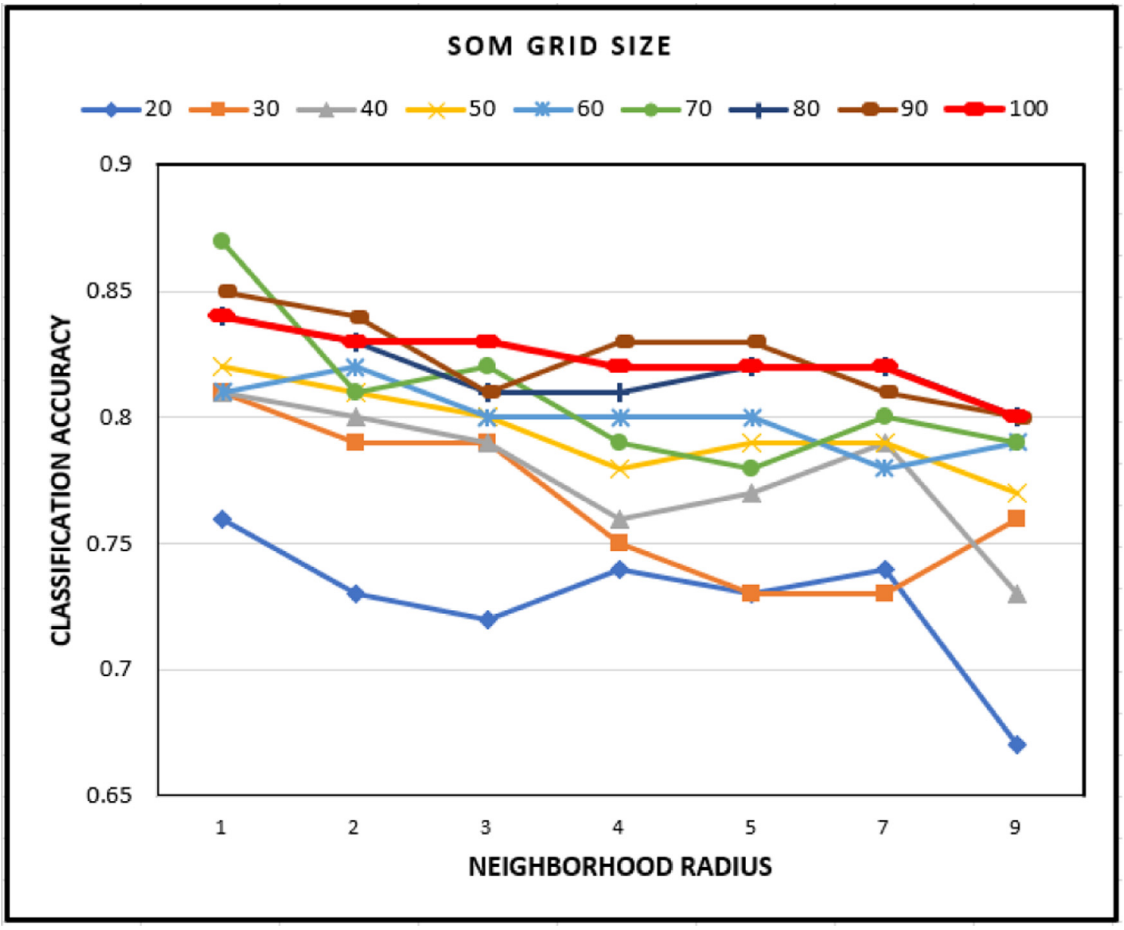

Fig. 10 – Effect of the SOM grid size and neighborhood radius on the performance using Word2Vec, 10-fold cross validated.

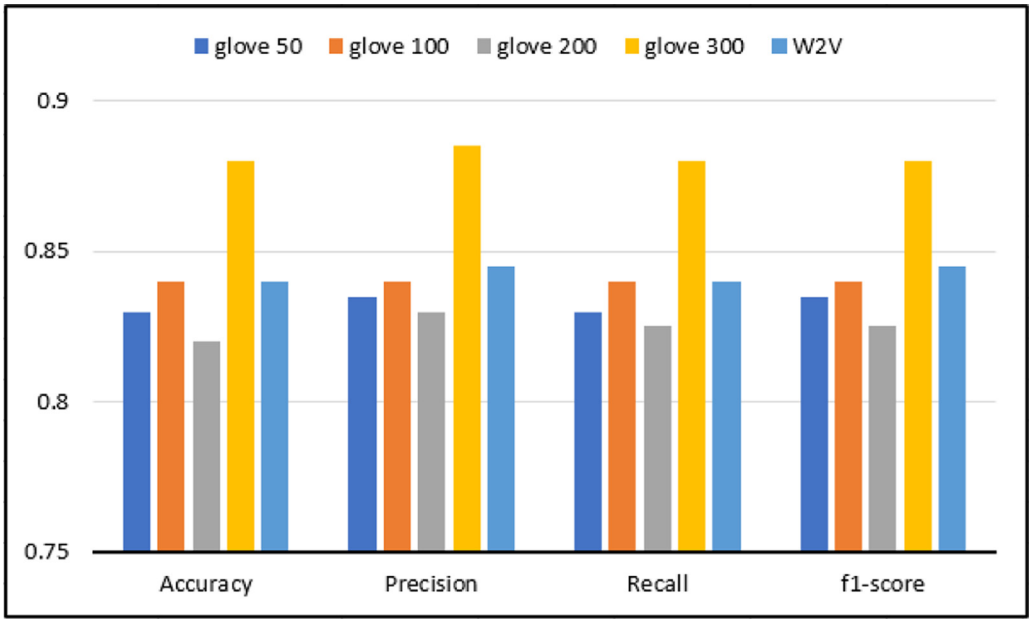

Fig. 11 – Effect of embedding method on the performance metrics.

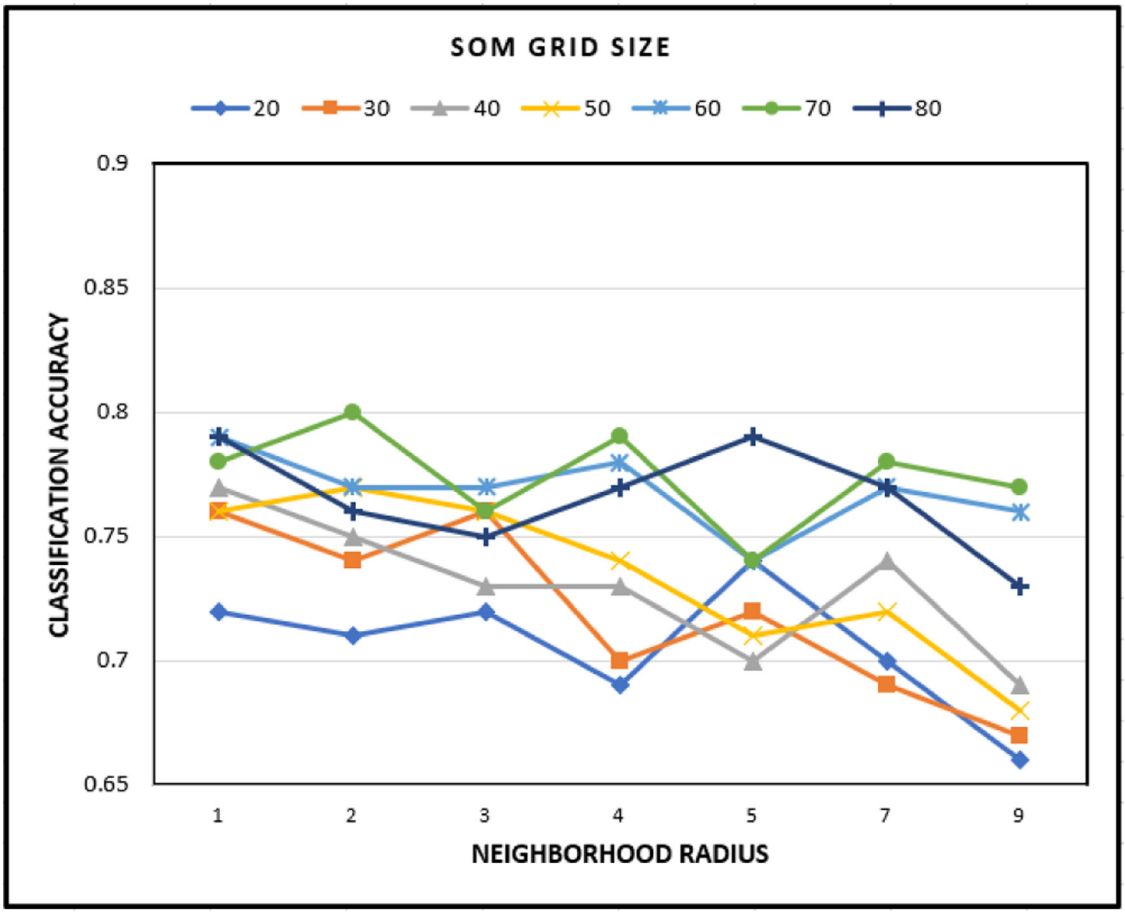

Fig. 12 – Effect of the SOM grid size and neighborhood radii on the performance on the Multi-domain dataset.

For the Multi-domain dataset (Yafeng and Donghong, 2017) used a neural network model to learn document-level representation for detecting deceptive opinion spam. First, they used a CNN to learn sentence representations. Then a gated recurrent neural network is utilized to combine those sentence representations to yield a document vector. Finally, the document representations are fed to various classifiers as features to identify deceptive opinion spam. They reported the performance of various neural network-based methods and compared them with their proposed method, Bi-directional Average GRNN. They used AMTs and customer reviews and split the data into 80 percent for training and 20 percent for testing and validation. Here, we provide their results and compared them with those of our method, SOM-CNN, in Table 5. To make a fair comparison, we used the same types and number of reviews, as mentioned above. The results show that our SOM-CNN model yields better accuracy than the best performing methods for spam review classification, on Multi-domain contexts.

Table 5 – Performance comparison of different methods for the multi-domain dataset.

Method	Accuracy	Macro-F1
Average	0.730	0.739
CNN	0.759	0.774
RNN	0.632	0.648
GRNN	0.790	0.799
Average GRNN	0.801	0.807
Bi-directional average GRNN	0.836	0.834
Le and Mikolov [2014]	0.761	0.776
SOM-CNN	0.871	0.870

As a recent work on spam review detection, Shahariar et al., 2019 used two sets of datasets, Ott and Yelp datasets. The Ott dataset is the same Single-domain dataset that we have also

Table 4 – Performance metrics for SOM-CNN on the multi-domain dataset by applying five-fold cross-validation.

	Accuracy	Precision	Recall	F1-score
Hotel	0.82	0.82	0.82	0.82
Doctor	0.85	0.84	0.83	0.83
Restaurant	0.83	0.83	0.83	0.83
Multi-domain	0.80	0.80	0.79	0.80

Table 6 – Performance comparison of single domain dataset.

Method	Trin test ratio	Accuracy
CNN (Shahariar et al., 2019)	90:10	95.56%
LSTM (Shahariar et al., 2019)	80:20	96.75%
MLP (Shahariar et al., 2019)	5-Fold	93.19%
CNN-SOM	10-Fold	87.63%

used in this work. They compared their own implementation of traditional state-of-the-art classifiers with NN-based methods of Multi-Layer Perceptron (MLP), CNN, and Long Short-Term Memory (LSTM) models. Since the Yelp dataset is not labeled, they used a modified active learning method to label the data. They used TD-IDF for MLP and Word2Vec embedding for CNN and LSTM to represent review texts as numerical values. The work is interesting as the authors could compare their own implementation of different methods. Table 6 provides a comparison for the Single-domain Ott dataset and shows that our cross validated results are comparable to theirs. Comparing to our work however, we noticed the following differences:

First, there is not much details on how they implemented the methods used. More related to direct comparison to our method, it is not stated how the variable length reviews were used as input to CNN classifiers that accept fixed length vectors as their inputs. In our approach, we have similarly used word embedding to convert all the words in the vocabulary into word vectors. Then, by introducing a novel approach using SOM we convert the variable length reviews to fixed size matrices (images), which can be fed into any classifier as proper inputs.

Secondly, accuracy is the only performance metric provided in their work, while we have provided many performance metrics validating the method performance. Our proposed method also showed working well on both single and multi domain data. Overall, through providing detail implementation, source code, verified and cross validated performance measures, we find our results more accurate, verifiable and valid in comparison.

6. Conclusion and future work

We have introduced a new method used to distinguish spam reviews from genuine ones. The proposed framework has been applied on a well-known dataset taking into account contextual features from the body of the reviews. We used two different embedding techniques to investigate the effectiveness of our method and compare their performance. We combined unsupervised learning via a SOM to cluster semantically-similar words and create images for the reviews. The review images are then fed to a CNN for training and classification.

The results show that the model based on GloVe-300 embedding achieves the best performance. Our SOM-CNN method yields promising results on both single and multi-domain contexts. A careful observation and visual inspection of the results reveals that the performance of our method has a direct relation to the size of the SOM grid map and the neighborhood radii. We used two features, word density and TF-IDF, in the map cells as two layers of the review images being constructed.

As future work, the performance of the proposed method could be further improved in the future by adding more layers of features to the review images created via SOMs. The proposed method can also be tested on other datasets and cross-domain reviews to investigate its effectiveness. Other types of features such as reviewer and metadata-based features can also be compiled and added to our model to boost its performance as reported in the literature for other methods.

Source code availability

The source code and sample data are available via a Github project at https://github.com/Ashsari/spam_review_detection. Since the project will stay private until publication of this paper, the source code is attached in a zip file for review purposes.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Ashraf Neisari: Data curation, Writing - original draft, Methodology, Software, Validation. **Luis Rueda:** Supervision, Investigation, Formal analysis, Conceptualization, Methodology, Resources, Funding acquisition, Writing - review & editing. **Sherif Saad:** Supervision, Investigation, Conceptualization, Methodology, Funding acquisition, Writing - review & editing.

Acknowledgments

This research work has been partially supported by the Natural Sciences and Engineering Research Council of Canada, NSERC, Grant No. RGPIN-2019-04696. The authors would like to thank the University of Windsor Office of Research Services and Innovation.

REFERENCES

Asghar N. Yelp Dataset Challenge: Review Rating Prediction. Ithaca: Cornell University Library; 2016. arXiv.org http://search.proquest.com/docview/2079845554/

Cardoso EF, Silva RM, Almeida TA. Towards automatic filtering of fake reviews. Neurocomputing 2018;309:106–16.

Duhan N, Divya, Mittal M. Opinion mining using ontological spam detection. In: 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS); 2017. p. 557–62.

Fatima N, Rueda L. iSOM-GSN: an integrative approach for transforming multi-omic data into gene similarity networks via self-organizing maps. Bioinformatics 2020;36(15):1367–4803.
Fei G, Mukherjee A, Liu B, Hsu M, Castellanos M, Ghosh R. Exploiting burstiness in reviews for review spammer detection. Proceedings of the International AAAI Conference on Web and Social Media. PKP Publishing Services Network, 2013.
Ho-Dac NN, Carson SJ, Moore WL. The effects of positive and negative online customer reviews: do brand strength and category maturity matter? J. Mark. 2013;77(6):37–53.
Hussain N, Mirza HT, Hussain I, Iqbal F, Memon I. Spam review detection using the linguistic and spammer behavioral methods. IEEE Access 2020;8:53801–16.
Jindal N, Liu B. Analyzing and detecting review spam. In: Seventh IEEE International Conference on Data Mining (ICDM 2007). IEEE; 2007. p. 547–52.
Kohonen T, Mäkisara K. The self-organizing feature maps. Phys. Scr. 1989;39(1):168–72.
Li J, Ott M, Cardie C, Hovy E. Towards a general rule for identifying deceptive opinion spam. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2014. p. 1566–76.
Liu Y, Pang B. A unified framework for detecting author spamicity by modeling review deviation. Expert Syst. Appl. 2018;112:148–55. doi:10.1016/j.eswa.2018.06.028.
Li FH, Huang M, Yang Y, Zhu X. 2011. Learning to identify review spam. In twenty-second international joint conference on artificial intelligence.
Shahariar G, Biswas S, Omar F, Shah FM, Hassan SB. 2019. Spam review detection using deep learning. In 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 0027-0033. IEEE, 2019.
Mukherjee, A, Kumar A, Liu B, Wang J, Hsu M, Castellanos M, Ghosh R. 2013. Spotting opinion spammers using behavioral footprints. In proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 632-640. 2013.
Fusilier Donato Hernández, Rafael Guzmán Cabrera, Manuel Montes, and Paolo Rosso. 2013. Using PU-learning to detect deceptive opinion spam. In proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis, pp. 38-45. 2013.
Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. 2013. Distributed representations of words and phrases and their compositionality. arXiv preprint arXiv:1310.4546
Ott M, Cardie C, Hancock JT. Negative deceptive opinion spam. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2013. p. 497–501.
Ott M, Choi Y, Cardie C, Hancock JT. Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics; 2011. p. 309–19.
Pennington J, Socher R, Manning C. GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar; 2014. p. 1532–43. doi:10.3115/v1/D14-1162. https://www.aclweb.org/anthology/D14-1162
Saini S, Saumya S, Prakash Singh J. Sequential purchase recommendation system for e-commerce sites. In: IFIP International Conference on Computer Information Systems and Industria Management; 2017. p. 366–75.
Saumya S, Singh JP. Detection of spam reviews: a sentiment analysis approach. Csi Trans. ICT 2018;6(2):137–48.
Saumya S, Singh JP, Baabdullah AM, Rana N, Dwivedi YK. Ranking online consumer reviews. Electron. Commer. Res. Appl. 2018;29:78–89. doi:10.1016/j.elerap.2018.03.008.
Rayana Shebuti, and Leman Akoglu. 2015. Collective opinion spam detection: bridging review networks and metadata. In Proceedings of the 21th ACN SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 985-994. 2015.
Singh JP, Irani S, Rana NP, Roy PK, Dwivedi YK, Saumya S. Predicting the helpfulness of online consumer reviews. Bus. Res. 2017(70):346–55.
Vesanto J, Alhoniemi E. Clustering of the self-organizing map. IEEE Trans. Neural Netw. 2000;11(3):586–600.
Vettigli G. 2020. MiniSom 1.1.2. https://test.pypi.org/project/MiniSom/. Accessed: May 30, 2020.
Wurtz RH. Recounting the impact of Hubel and Wiesel. J. Physiol. 2009;587(12):2817–23.
Yafeng R, Donghong J. Neural networks for deceptive opinion spam detection: an empirical study. Inf. Sci. 2017;385:213–24.
Yilmaz CM, Durahim AO. SPR2EP: a semi-supervised spam review detection framework. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM); 2018. p. 306–13.
Zhu F, Zhang X. Impact of online consumer reviews on sales: the moderating role of product and consumer characteristics. J. Mark. 2010;74(2):133–48.

Ashraf Neisari graduated in Computer Science and received her bachelor's degree from Azad University, Tehran, Iran, in 2012. She received the Computer Networking Technology Diploma from St. Clair College, Windsor, Ontario, Canada and got her CompTIA Security + certificate in 2017. She was a sessional instructor at St. Clair College teaching Computer Networking undergraduate courses Networking III and Networking IV and ran the lab for those courses. She received her Master's degree in Computer Science in 2020 from the University of Windsor, Ontario, Canada under the supervision of Professors Luis Rueda and Sherif Saad. Her research interests are focused on machine learning, networking and cybersecurity. Luis Rueda received his Bachelor's degree in computer science from the National University of San Juan, Argentina, in 1993, and his Master's and Ph.D. degrees in computer science from Carleton University, Canada, in 1998 and 2002, respectively. He is currently a Full Professor in the School of Computer Science at the University of Windsor. His current research interests are mainly focused on devising shallow and deep machine learning algorithms at the fundamental level and applications to protein-protein interaction, transcriptomics, next generation sequencing, integrative genome-wide analysis, cybersecurity, social engineering and identification of cancer biomarkers.Luis Rueda holds four patents on data encryption and has more than 200 publications and presentations in prestigious journals and conferences in machine learning, computational biology and data security. He currently serves as Associate Editor of IEEE/ACM Transactions on Computational Biology and Bioinformatics. He is also a member of the Technical Committee on Pattern Recognition for Bioinformatics (IAPR TC-20) and the program committees of several conferences in the field. He is also a Senior Member of the IEEE, and a Member of the Association for Computing Machinery and the International Society for Computational Biology. **Sherif Saad** is an assistant professor of cybersecurity at the University of Windsor and found the WASP (Windsor Advanced Security and Privacy) research lab that develops innovative and usable security solutions for unconventional cybersecurity threats. His research interests include cybersecurity, applied machine learning, and software engineering. With WASP labs, Dr. Saad is leading several research projects with NRC, Canada DND, NSERC, MITACS, and other

private organizations. Dr.Saad has published several articles in prestigious and top-tier computing and cybersecurity journals and conferences. Dr. Saad has 10+ years of industry experience in cybersecurity and applied machine learning. During these years, he had the following roles: software developer, application security engineer, software security architect, chief software architect, and director of engineering. He worked with many companies to develop security systems for clients in the defence and finance sectors. Some clients include MasterCard, American Express, US DoD, Booz Allen, RCMP, and DRDC.