Conical Classification For Computationally Efficient One-Class Topic Determination

Sameer Khanna
Department of Computer Science, Stanford University, USA
Department of Research and Development, Fortinet, USA
sameerk@stanford.edu

Abstract

As the Internet grows in size, so does the amount of text based information that exists. For many application spaces it is paramount to isolate and identify texts that relate to a particular topic. While one-class classification would be ideal for such analysis, there is a relative lack of research regarding efficient approaches with high predictive power. By noting that the range of documents we wish to identify can be represented as positive linear combinations of the Vector Space Model representing our text, we propose Conical classification, an approach that allows us to identify if a document is of a particular topic in a computationally efficient manner. We also propose Normal Exclusion, a modified version of Bi-Normal Separation that makes it more suitable within the one-class classification context. We show in our analysis that our approach not only has higher predictive power on our datasets, but is also faster to compute.

1 Introduction

In the era of the rapid development of computers and the Internet, information on a wide range of topics is pervasive. The amount of text based data is ever increasing in size, magnitude, and variety. Whether it is for e-commerce (Xiao and Tong, 2021), clinical diagnosis determination (Le et al., 2021), or fake news detection (Ahmed et al., 2018) it is vital to have efficient mechanisms for topic classification in order to effectively parse and process text based media.

Most of the research on topic classification uses these implementations within a binary classification or multi-class classification context (Trstenjak et al., 2014; Zhang et al., 2011; Kim and Gil, 2019; Kim et al., 2019a; Liu et al., 2018). Comparatively, there is a relative dearth of content variety discussing and proposing different algorithms that can identify text on a particular subject from a variety of subjects in a One-Vs-All configuration, especially regarding how to use vector representations of documents with low computational costs. This is unfortunate, as one class classification of text enables us to identify text of a particular form from a potentially non-exhaustible set of potential topics. In such a setting, it would be arduous to identify all potential topics we may come across and extremely time-consuming to label enough data to train a model for multi-class classification.

In practice, the lack of research into one class topic determination has lead to subpar implementations for the sake of speed. One of the best examples of the ramifications of this lack of research focus is insider threat detection systems. Despite insider threat detection primarily working with log and textual information, the vast majority of published work on the subject do not utilize Natural Language Processing in their implementations (Wei et al., 2021; Tuor et al., 2017; Meng et al., 2018; Le et al., 2018; Le and Zincir-Heywood, 2019). Many that do simply sum over TF-IDF vectors before feeding the result as a feature into detection models (Chattopadhyay et al., 2018; Sajjanhar et al., 2019).

We aim to tackle these issues head on. Our contributions are as follows:

- We propose Normal Exclusion, a re-framing of Bi-Normal Separation enabling usage for one-class classification.
- We show that our approach, Conical Classification (CC), achieves optimal performance when compared to alternative one-class topic determination strategies.

2 Related Work

With the intention of assessing the predictive power of one-class based text classification methods, Joffe et al. has compared one-class support vector machines (OCSVM) to binary support vector machines (SVM) to identify specific phenotypes in

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 1662–1673
November 7–11, 2021.

breast cancer. They found that OCSVM performed comparably to SVM in balanced dataset problem spaces and outperformed SVM in highly imbalanced datasets (Joffe et al., 2015).

Zhuang et al. concurs, citing the improved performance of switching from a SVM to OCSVM approach for minority class classification. They use a general framework which first uses the minority class for training in the one-class classification stage, then incorporate data from the majority class to improve the generalization performance of the constructed classifier (Zhuang and Dai, 2006).

It turns out that OCSVMs have a wide adoption rate for one-class text classification problems. Additional examples include Manevitz et al. using them for document classification (Manevitz and Yousef, 2001), and Seo utilizing a OCSVM to help classify images in a database using color and text content for content-based image retrieval (Seo, 2007).

Ensemble based methodologies have been used in practice as well. Hempstalk et al. has utilized an ensemble-based approach using C4.5 decision trees with Laplace smoothing to isolate real target values from those of an artificial class (Hempstalk et al., 2008), validating performance on various UCI datasets as well as a custom typist dataset. Anderka et al. utilized a similar approach to detect text quality flaws, using a Random Forest as the base classifier instead (Anderka et al., 2011). Unfortunately, despite their higher memory and computation requirements, such approaches have little performance benefits compared to the OCSVM; Hempstalk et al.'s results indicated that their ensemble approach was not demonstratively superior to the OCSVM approach.

While not traditional one-class classification algorithms, there are a set of classifiers that co-train using a set of positive labeled data as well as a set of unlabeled data for evaluation. Denis et al. has developed the Positive Naive Bayes (PNB) classifier that works under this setting, using it successfully to classify documents in the 20-Newsgroup dataset (Denis et al., 2003).

One-class topic determination is a problem space where it is paramount to be computationally fast with low resources in order to process large numbers of documents in a short amount of time. This has traditionally excluded recent advancements in Natural Language Processing such as embeddings from the discussion, as these take significant amounts of computation time on the modest hardware such application spaces necessitate. This has resulted in very few publications dedicated to assessing their application to the space. Ruff et al. propose Context Vector Data Description (CVDD) (Ruff et al., 2019), a textual anomaly detection algorithm that builds upon word embedding models to learn multiple sentence representations that capture multiple semantic contexts via the self-attention mechanism. Hu et al. extended uni-modal Support Vector Data Description (SVDD) to a multiple modal one, building Multi-modal Deep Support Vector Data Description (mSVDD) with multiple hyperspheres, enabling them to build better descriptions for target one-class data (Hu et al., 2021).

The methodology used to create the vector representations of documents can be just as important as the detection algorithm used. One main approach that has come about as a result is term frequency (TF) – inverse document frequency (IDF). TF-IDF is the product of two statistics: TF and IDF. TF, as its name suggests, refers to the normalized frequency f of a word w_j that appears in the given document D. Originally coined as term specificity by Jones (Jones, 1972), IDF provides a measure of how much information a word provides depending on how common the word is in a given corpus.

TF-IDF has been successfully used for topic classification in a variety of scenarios, ranging from social media (Lee et al., 2011), research analysis (Kim and Gil, 2019), and news discovery (Hakim et al., 2014). As a result, much research has been done on modifications to improve performance. Martineau et. al. has proposed Delta TF-IDF which scales weights using word scores before classification and boasts a higher accuracy than standard TF-IDF (Martineau and Finin, 2009). Forman studies replacing TF-IDF with Bi-Normal Separation (BNS), eliminating the need for fine-tuned feature selection and performs exceptionally well on short length documents (Forman et al., 2003). Domeniconi et. al. used a supervised variant to prevent the IDF term from affecting documents within the category under analysis, so that terms frequently appearing in said category are not penalized (Domeniconi et al., 2015).

More recently, vector representations have been developed that use embeddings, such as BERT (Devlin et al., 2018) and GloVe (Pennington et al., 2014). Such embeddings allow for words with similar meanings to have a similar representation

which has allowed for the impressive performance of deep learning methods on complex and intricate natural language processing problem spaces.

3 Normal Exclusion

BNS, which is the measure of how much the probability of occurrence of a given word in the positive class differs from the probability of occurrence of a given word in the negative class, has a couple of key benefits as a VSM metric: it is excellent at ranking words for automated feature selection filtering, it has the best performance in single metric VSM analyses, and is consistently a member of the optimal pairs of VSM metrics Forman et al. evaluated (Forman, 2008). Thus, being able to utilize BNS within a one-class context would be ideal.

The formula used to calculate BNS is given in Equation 1. Here, tpr is the true positive rate $P(word|positiveclass)$ as determined via $\frac{tp}{pos}$, where tp is the number of positive training cases containing the word and pos is the number of positive training cases. Likewise, fpr refers to the false positive rate $P(word|negativeclass)$ as determined via $\frac{fp}{neg}$, where fp is the number of negative training cases containing word and neg is the number of negative training cases. F^{-1} is the inverse Normal cumulative distribution function. ϵ is a number with small magnitude added to avoid the undefined scenario of $F^{-1}(0)$; for the purposes of our analysis, we set ϵ to 0.0005, or half a count out of 1000.

$$\text{BNS} = \left|F^{-1}(tpr + \epsilon) - F^{-1}(fpr + \epsilon)\right| \quad (1)$$

A naive translation to a one-class regimen would be to merely remove BNS's dependence on the fpr term. Thus, each word would be scaled in relation to its frequency of occurrence within our positive training set. This leads to issues, as words with a naturally high occurrence in language such as the, be, to, of, a, etc. will have predominantly high scaled values. One may try to work around these effects by removing stopwords and unrelated words from our corpus, but this can require significant hand-tuning by an expert in the field while increasing overhead computation costs.

We propose an alternative solution that takes advantage of the nature of one-class classification, recalling that we wish to be able to identify text of a particular topic from any assortment of topics possible from the language. We simply need to estimate the fpr of the word with the frequency of the word in our given language. For English, there are large corpuses from which we can extract this information, for example the Oxford English Corpus (OEC) is a dataset that presents all types of English, from blogs to newspaper articles to literary novels and even social media, sourcing from Englishes from the United Kingdom, the United States, Ireland, Australia, New Zealand, the Caribbean, Canada, India, Singapore, and South Africa. For our purposes, we compiled the frequencies of the top $\frac{1}{3}$ million words in the human language using Tatman's English word count dataset (Tatman, 2017; Brants and Franz, 2006) and stored them within a dictionary for rapid lookup.

We can safely set the frequencies of words that do not appear in our dictionary to 0, as these include words that rarely appear in standard language; such words include abaptiston, abaxile, grithbreach, gurhofite, zarnich, and zeagonite. Indeed, according to Oxford's compiled statistics, the combined frequency of occurence for all such words is approximately a percent of the entire lexicon of the English language, easily within the margin of error for our analysis (Oxford, 2011).

$$\text{NE} = \left|F^{-1}(tpr + \epsilon) - F^{-1}(Dict[word] + \epsilon)\right| \quad (2)$$

We coin our tweaked formula Normal Exclusion (NE), as it excludes, or reduces, the weightage of words that are inconsequential to determining the topic of text without requiring a negative corpus to be present. The formula for NE is shown in Equation 2. Here, Dict[word] represents the frequency value for the given word as found within our dictionary.

We will scale NE by TF for our model developing the NE-TF VSM. Our representation of a word in a model will thus be determined by how frequently a word occurs in our corpus, scaled by the statistical significance of the word within the evaluated text. Higher magnitude values give a strong indication that the vector is about our target topic, while lower values would lead to a lower confidence that such a conclusion is correct.

4 Conical Classification

4.1 Why Positive Span

VSM is based on the notion of vector similarity; the model assumes that the relevance of a document to another document is roughly equal to the document-query similarity. Under this model, the documents are represented using the *bag-of-words*

approach. This means that documents are translated to n-dimensional vectors, where each dimension corresponds to a word based on a compiled set of terms known as a vocabulary. Under such models, we map a given topic to a certain subset of the compiled vocabulary.

It is not enough however for a document to have a high frequency of words included within the subset to be classified as a given topic. Combinations of words are vital to the classification process. For a timely example, a news article regarding COVID-19 and an administration protocol manual on COVID-19 vaccines will both strongly correlate to words such as vaccines, dosages, Pfizer, Moderna, among others. To distinguish between these two topics, we would need contextual words such as policy, mandate, and president to identify a news article, and words like intramuscular, angle, deltoid, and subcutaneous would likely exist within an administration protocol manual. While these contextual words will have a lower correlation to a given topic, they are nonetheless paramount for an effective classification model. This leads to a high significance of vector orientation within a VSM as it is crucial to keep track of how a word represented by a certain dimension relates to words represented by different dimensions.

The high interdependence between VSMs and orientation allows one to assess document similarity solely from the context of vector angles. For example, to rank similarity within a category, a simple and popular mechanism is to calculate the Relevance Status Value which computes the cosine of the angle between the query and each document in the collection (Rao and Gudivada, 2018). The larger the cosine value, the smaller the angle, and the more similar the documents being compared are. It is important to note at this point that while vector magnitude would typically be a crucial metric to consider as well, Rao et al. furthers, stating VSM vectors are typically normalized before further computation and analysis is done.

This means that documents of the same topic will have smaller angles between each other than those comprised of different topics altogether. Extrapolating from this observation to the comparison of a document to an entire corpus, we expect for vectors corresponding to the same topic to be close to the center of the distribution of corpus vectors in order to have a low angle to all vectors in the corpus. Similarly, we expect vectors from a different topic to have a high angle from the vectors in the corpus. Figure 1 provides an illustration of the expected phenomenon.

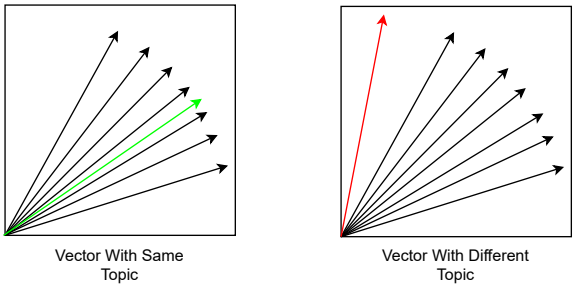

Figure 1: New document vector of the same topic versus new document vector of a different topic. Green refers to a document that will be classified as having the same topic, red will be classified as not having the same topic.

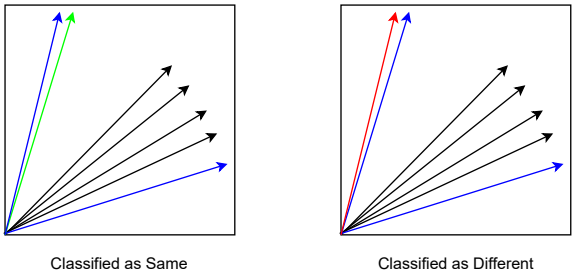

Figure 2: Edge cases for our classification system. Green and red remain as defined in Figure 1. Blue refers to our corpus fringe vectors.

Note we do not yet consider documents that are edge case scenarios. To simplify nomenclature for further discussion, we refer to vectors within our corpus that are most dissimilar to the other vectors in the corpus our fringe vectors. We consider fringe vectors to be as distant from the corpus as possible while still being considered as having the same topic. Thus, as shown in Figure 2, the similarity with respect to a fringe vector is not sufficient to be classified as having the same topic as the given corpus; if a vector is similar to a fringe vector, but less similar to rest of the corpus than the fringe vector, we will consider the vector being evaluated to be of a different topic. In other words, a vector must be in-between our fringe vectors across all dimensions to be considered as having the same topic as our corpus.

From here, we can translate the classification problem into a linear combination problem. As shown in Figure 3 for the two dimensional case, any vector found in between two vectors can be

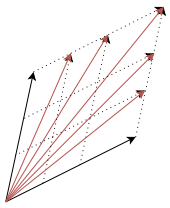

Figure 3: Any vector found between two vectors can be created from the linear combination of its surrounding vectors.

represented by their linear combination. We define a vector as being in-between two vectors if the sum of its angles to each vector is equal to the angle between the two vectors themselves and it lies on the plane defined by the two vectors. Note this vector can always be calculated as a linear combination of its surrounding vectors; Algorithm 1 shows an approach based on binary search that allows one to identify the scalar combinations needed to recreate the target vector. Here, cos_{sim} refers to cosine similarity (Sitikhu et al., 2019), $target$ is the vector we are trying to recreate, x and y are the vectors $target$ is in-between while λ_x and λ_y are the scalar values such that $x\lambda_x + y\lambda_y = target$.

Algorithm 1: Binary Search Approach To Finding Linear Combination Scalars For Target Vector In-between Two Vectors

Result: λ_x, λ_y
$vector_{one} = x$;
$vector_{two} = y$;
$mid = \frac{vector_{one}+vector_{one}}{2}$;
$\lambda_x = \frac{1}{2}$;
$\lambda_y = \frac{1}{2}$;
$level = 1$;
while $mid \neq target$ **do**
  $level = level + 1$;
  $sim_{one} = cos_{sim}(vector_{one}, target)$;
  $sim_{two} = cos_{sim}(vector_{two}, target)$;
  if $sim_{one} \geq sim_{two}$ **then**
    $mid = vector_{two}$;
    $\lambda_x = \lambda_x + 2^{-level}$;
    $\lambda_y = \lambda_y - 2^{-level}$;
  else
    $mid = vector_{one}$;
    $\lambda_x = \lambda_x - 2^{-level}$;
    $\lambda_y = \lambda_y + 2^{-level}$;
  end
end

This conclusion also makes intuitive sense. As discussed earlier, we can identify a document as being from a particular topic if it has word combinations that indicate as such. A vector that is a linear combination of those within the corpus must have one or more such identifying word combinations as a result.

It is important to note that by linear combinations, we specifically refer to the set of positive linear combinations. As mentioned earlier, orientation of vectors is crucial in regards to which documents and word combinations they represent. A negatively scaled vector represents the complete opposite document than a positively scaled counterpart and thus should not be used for topic classification.

We have shown it is enough to compose a vector as a positive linear combination of the vectors in a corpus to confirm that it is regarding a similar topic. In other words, a document has the same topic as a corpus if its vector representation is within the positive span of the corpus.

4.2 Conical Sets

The positive span of vectors v_1 through $v_k \in \mathbb{R}^n$ is the linear combination $\sum_i^k \lambda_i v_i$ where $\lambda_i \geq 0$ for all i = 1, ..., k (Davis, 1954). Note that the original definition made by Davis allows for the zero vector to be included within the positive span. However, the zero vector within the VSM context represents a vector with none of the terms corresponding to the corpus topic; we thus wish to exclude the zero vector from our span in order to properly classify documents based on their topic. Our new span, coined the conical span, of vectors v_1 through $v_k \in \mathbb{R}^n$ is the linear combination $\sum_i^k \lambda_i v_i$ where $\sum_i^k \lambda_i > 0$ for all $i = 1, ..., k$.

We define the conical set that can be defined via the conical span of a finite number of vectors in Equation 3.

$$\text{conical(S)} := \{\lambda_1 v_1 + ... + \lambda_k v_k : \lambda_1 + ... + \lambda_k > 0\} \quad (3)$$

Conical span enables a large range of possibilities from the positive span of vectors; Figure 4 showcases the vast representational power in three dimensional space, where the addition of extra vectors dramatically increases the variety of subspace shapes that can be created (Stappen, 2020).

4.3 Two Vector Comparison

At this point, we have shown that it is sufficient for topic classification that a vector is within the conical span, and we have displayed the expressive power of the conical span. We will now go over an efficient mechanism to determine if a vector is within the conical span. As Rao et al. claims is standard for VSM vectors, we assume all vectors

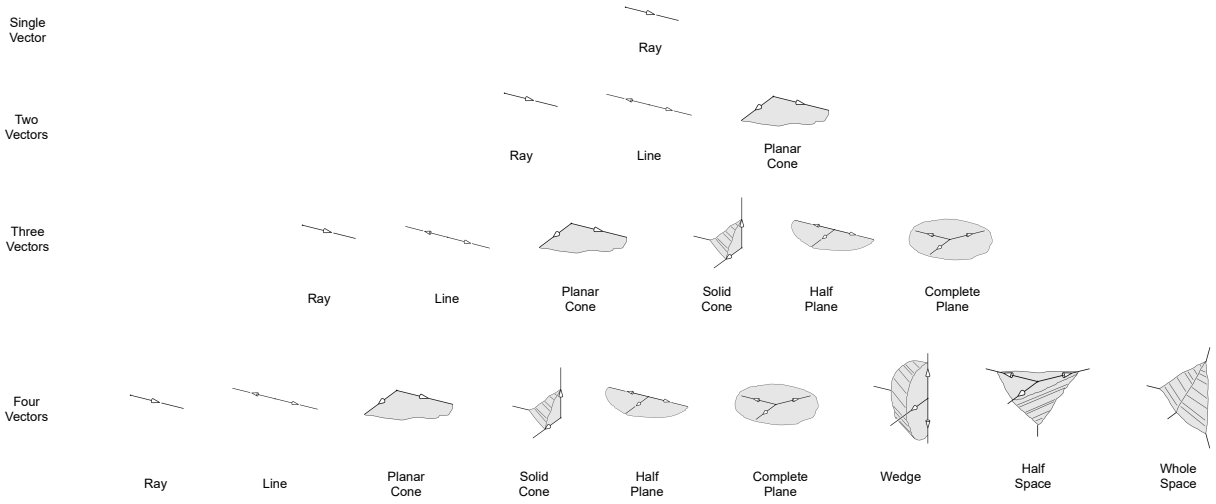

Figure 4: Conical set subspaces comprised of different vector totals in three dimensional space.

in our corpus as well as our evaluation vectors are unit norm in length (Rao and Gudivada, 2018).

In order to train our CC system, we simply find the largest value and the smallest value for every dimension in our corpus vectors, and store them within two vectors for analysis later on. Then when it comes to predicting with CC, we simply need to compare our evaluation vector with both vectors in order to determine if the vector belongs to our corpus.

We prove this claim via the following lemmas and theorems.

Lemma 4.1. *There is no unit vector within the conical span that is larger in one or more dimensions than the max vector or smaller in one or more dimensions than the min vector.*

Proof. We prove by contradiction. Assume there is a vector in the conical span whose value in one or more dimensions is larger than than the max vector or smaller than the min vector. The values in the max and min vectors are set by the fringe vectors for the given dimensions, due to these vectors having the largest deviation from the corpus acceptable. For a vector to have a value outside of this range, the given vector must deviate further from the rest of the corpus than our fringe vectors. This leads to a contradiction; by definition, any vector less similar to our corpus than the fringe vectors must not be classified as being of the same topic and thus within the conical span. □

Lemma 4.2. *There is no unit vector outside the conical span that is smaller in a given dimension than the max vector and larger in a given dimension than than the min vector.*

Proof. We prove by contradiction. Assume that a unit length vector outside the conical span exists such that its values are in between the min and max vectors. As mentioned in Lemma 4.1, the max and min vectors are defined by the value of our fringe vectors for each dimension of our VSM. A value closer in similarity to the rest of our corpus is by definition within our conical span. This leads to a contradiction: a vector cannot both be more similar to main vectors within our corpus than our fringe vectors and be classified as a different topic. □

Theorem 4.3. *All possible unit vectors in the conical span can be represented by a max and min vector.*

Proof. By combining Lemmas 4.1 and 4.2, we arrive at the conclusion that Theorem 4.3 is indeed correct. □

This result enables us to rapidly train and determine if a given vector is of a certain topic or not. If a vector is not the zero vector, is less than the max vector across all dimensions, and is greater than the min vector across all dimensions, then we classify the vector as being of the same topic as it is within the conical span of the topic training corpus.

5 Evaluation Methodology

5.1 Baseline Models

As detailed in the Related Works section, OCSVMs have an extremely high adoption rate within the

space, thus for our analysis, we evaluate the performance of OCSVMs on the following kernel functions: linear, sigmoid, radial basis function (RBF), and polynomial (Poly).

To represent our set of ensembles, we will train a One Class Random Forest classifier (OCRF) using Goix et al.'s splitting method (Goix et al., 2017) as well as an Isolation Forest classifier (IsolFor) (Liu et al., 2008). For both methods, we will use 1000 estimators.

We also utilize PNB as a baseline measure. Since we wish to evaluate its performance in the one-class classification regime, we will use the evaluation data itself as the unlabeled set of data for training the algorithm; this allows us to only pass in the positive set of data points during training as is the case for traditional one-class classification algorithms.

Finally, to represent embedding based models, we use CVDD as our representation for context preserving embedding based approaches as well as for neural NLP, taking advantage of the official implementation known as CVDD-PyTorch (Ruff et al., 2019). Both GloVe and BERT models are assessed for evaluation purposes, with embedding size, attention size, and number of attention heads set to be the best performing configuration .

Except for our CVDD baselines, all of our baseline models will use TF-IDF as the VSM of choice.

5.2 Dataset

Our intent is to evaluate our baselines as well as CC in scenarios that can require high performance. As mentioned in the Introduction, one main place where this can occur is in insider threat detection. The golden standard dataset for insider threats is the CERT Insider Threat dataset, the largest public repository of insider threat scenarios compiled after analyzing 1,154 actual insider incidents (Glasser and Lindauer, 2013). Within this dataset, there are three key website topics that are crucial to detect: Keylogger websites, Wikileaks-like websites, and job posting sites. We extract the text related to both Keylogging and Wikileaks by hand-labeling the text content within version 4.2 in order to use them both for evaluation purposes.

For the purpose of evalauting the latter of the three, we extract text related information from Vidros et al.'s Fake JobPosting Prediction dataset (Vidros et al., 2017), and from PromptCloud's job dataset (PromptCloud, 2017). Both are high quality datasets listing full descriptions of jobs with large varieties, and versions of both datasets have been used by a plethora of publications (Balachander and Moh, 2018; Kim et al., 2019b; Alghamdi et al., 2019; Mahbub and Pardede, 2018; Reddy et al., 2018). For our purposes, we extract text data from the real job postings in Vidros et al.'s dataset.

We also desire our evaluation set to have exposure to e-commerce applications, medical record information, and fake news articles. For our ECommerce dataset, we utilize the Women's Clothing E-Commerce dataset (Agarap and Grafilon, 2018), which has seen popularity for sentiment analysis and text classification tasks (Sun et al., 2019; Lin, 2020; Kousta and Bellet; Cascaro et al., 2019). Our MedicalTranscription dataset consists of text extracted from the Collection of Transcribed Medical Transcription Sample Reports and Examples (MTSamples), a dataset of interest in academia both from a natural language processing perspective as well as from a medical assessment point of view (Beattie and Richards, 1994; Moramarco et al., 2021; Zuccon et al., 2014). Finally, for FakeNews, we utilize the Fake and real news dataset (Ahmed et al., 2018, 2017); this dataset is especially relevant due to recent increases in the proliferation and rapid diffusion of fake news on the Internet.

We chose this set of classification markers not only due to its representation of some of the fields we expect CC to be applicable, but also due to the high variability in text length and composition; our Wikileaks and Keylogger datasets are small length texts composed primarily of keywords, whereas the MoviePlot and MedicalTranscription datasets have relatively verbose text covering complex and protean topic ranges. This large variety is crucial as research has shown that text length and topic variations have a dramatic affect on text-based classification performance (Wang and Manning, 2012).

5.3 Evaluation Setup

When a given dataset is being evaluated as the positive class, the rest of the datasets are combined and treated as the negative class. Since our training set does not require any data from the negative class, we split each class via a 50%-50% split between our validation and test sets. Our positive class is split using a 70%-15%-15% split between our training set, our validation set, and our test set. Resplitting our train and test sets each run, we compile evaluation metrics accuracy, balanced

Table 1: Performance metrics and computation times for all algorithms on evaluation datasets. Best results are bolded.

Dataset	Model	Accuracy	Balanced Accuracy	Precision	Recall	F1 Score	Time (s)
ECommerce	Linear OCSVM	0.900 ± 0.001	0.755 ± 0.003	0.978 ± 0.002	0.512 ± 0.007	0.672 ± 0.006	183.255 ± 3.821
	RBF OCSVM	0.899 ± 0.001	0.753 ± 0.004	0.981 ± 0.003	0.508 ± 0.008	0.669 ± 0.007	201.339 ± 3.792
	Sigmoid OCSVM	0.899 ± 0.002	0.752 ± 0.005	0.979 ± 0.003	0.508 ± 0.011	0.668 ± 0.009	193.402 ± 7.382
	Poly OCSVM	0.885 ± 0.006	0.712 ± 0.002	$\mathbf{0.995 \pm 0.001}$	0.424 ± 0.003	0.595 ± 0.003	183.813 ± 2.639
	IsolFor	0.199 ± 0.000	0.500 ± 0.000	0.199 ± 0.000	$\mathbf{1.000 \pm 0.000}$	0.333 ± 0.000	280.893 ± 12.543
	OCRF	0.953 ± 0.000	0.947 ± 0.0176	0.800 ± 0.000	0.898 ± 0.036	0.889 ± 0.000	189.252 ± 244.041
	PNB	0.638 ± 0.202	0.762 ± 0.092	0.786 ± 0.203	0.757 ± 0.305	0.687 ± 0.153	111.839 ± 59.581
	GloVe CVDD	0.948 ± 0.017	0.906 ± 0.026	0.951 ± 0.015	0.983 ± 0.007	0.967 ± 0.011	188.462 ± 12.773
	BERT CVDD	0.951 ± 0.128	0.910 ± 0.013	0.954 ± 0.017	0.987 ± 0.012	0.973 ± 0.028	233.626 ± 23.796
	CC	$\mathbf{0.988 \pm 0.009}$	$\mathbf{0.988 \pm 0.007}$	0.956 ± 0.005	0.988 ± 0.009	$\mathbf{0.971 \pm 0.002}$	$\mathbf{10.878 \pm 0.036}$
FakeNews	Linear OCSVM	0.818 ± 0.039	0.685 ± 0.082	0.819 ± 0.157	0.430 ± 0.232	0.495 ± 0.175	496.206 ± 5.353
	RBF OCSVM	0.813 ± 0.033	0.706 ± 0.065	0.772 ± 0.184	0.500 ± 0.229	0.538 ± 0.086	533.529 ± 11.002
	Sigmoid OCSVM	0.760 ± 0.046	0.648 ± 0.091	0.757 ± 0.259	0.436 ± 0.347	0.389 ± 0.166	514.913 ± 19.340
	Poly OCSVM	0.850 ± 0.014	0.722 ± 0.053	0.846 ± 0.071	0.476 ± 0.128	0.592 ± 0.091	470.485 ± 2.950
	IsolFor	0.238 ± 0.000	0.500 ± 0.000	0.238 ± 0.000	$\mathbf{1.000 \pm 0.000}$	0.384 ± 0.000	278.308 ± 5.449
	OCRF	0.930 ± 0.000	0.955 ± 0.000	0.761 ± 0.000	1.000 ± 0.000	0.864 ± 0.000	197.232 ± 255.352
	PNB	0.624 ± 0.223	0.773 ± 0.097	0.900 ± 0.142	0.697 ± 0.283	0.729 ± 0.152	218.296 ± 38.515
	GloVe CVDD	0.906 ± 0.003	0.881 ± 0.011	0.938 ± 0.018	0.936 ± 0.022	0.936 ± 0.002	282.293 ± 31.88
	BERT CVDD	0.899 ± 0.121	0.878 ± 0.218	0.923 ± 0.342	0.933 ± 0.231	0.927 ± 0.153	322.513 ± 49.659
	CC	$\mathbf{0.985 \pm 0.005}$	$\mathbf{0.985 \pm 0.004}$	$\mathbf{0.955 \pm 0.002}$	0.987 ± 0.006	$\mathbf{0.969 \pm 0.001}$	$\mathbf{13.952 \pm 0.026}$
Jobs	Linear OCSVM	0.913 ± 0.001	0.768 ± 0.004	0.971 ± 0.002	0.540 ± 0.009	0.694 ± 0.007	368.263 ± 6.227
	RBF OCSVM	0.910 ± 0.001	0.758 ± 0.003	0.975 ± 0.002	0.520 ± 0.007	0.678 ± 0.006	408.096 ± 5.810
	Sigmoid OCSVM	0.912 ± 0.006	0.766 ± 0.001	0.968 ± 0.001	0.537 ± 0.003	0.690 ± 0.002	386.841 ± 15.345
	Poly OCSVM	0.903 ± 0.001	0.736 ± 0.002	$\mathbf{0.989 \pm 0.001}$	0.474 ± 0.005	0.641 ± 0.005	364.713 ± 6.367
	IsolFor	0.181 ± 0.000	0.500 ± 0.000	0.181 ± 0.000	$\mathbf{1.000 \pm 0.000}$	0.307 ± 0.000	291.792 ± 8.304
	OCRF	0.961 ± 0.000	0.976 ± 0.000	0.818 ± 0.000	1.000 ± 0.000	0.900 ± 0.000	196.165 ± 253.234
	PNB	0.585 ± 0.214	0.696 ± 0.154	0.684 ± 0.290	0.869 ± 0.214	0.690 ± 0.166	189.128 ± 58.950
	GloVe CVDD	0.896 ± 0.037	0.836 ± 0.056	0.910 ± 0.054	0.961 ± 0.033	0.933 ± 0.025	271.637 ± 30.758
	BERT CVDD	0.886 ± 0.023	0.831 ± 0.049	0.903 ± 0.034	0.958 ± 0.025	0.918 ± 0.012	316.066 ± 32.659
	CC	$\mathbf{0.995 \pm 0.017}$	$\mathbf{0.994 \pm 0.000}$	0.985 ± 0.006	0.993 ± 0.004	$\mathbf{0.988 \pm 0.004}$	$\mathbf{11.115 \pm 0.021}$
Keylogger	Linear OCSVM	0.999 ± 0.004	0.706 ± 0.041	1.000 ± 0.000	0.413 ± 0.081	0.580 ± 0.078	$\mathbf{10.330 \pm 1.035}$
	RBF OCSVM	0.999 ± 0.009	0.705 ± 0.080	1.000 ± 0.000	0.410 ± 0.161	0.564 ± 0.156	11.373 ± 0.366
	Sigmoid OCSVM	0.999 ± 0.001	0.705 ± 0.093	1.000 ± 0.000	0.411 ± 0.187	0.556 ± 0.192	10.606 ± 0.766
	Poly OCSVM	0.999 ± 0.007	0.745 ± 0.066	1.000 ± 0.000	0.491 ± 0.131	0.648 ± 0.124	10.361 ± 0.831
	IsolFor	0.791 ± 0.294	0.720 ± 0.031	0.666 ± 0.470	0.649 ± 0.251	0.428 ± 0.303	276.514 ± 22.813
	OCRF	1.000 ± 0.000	1.000 ± 0.000	0.999 ± 0.000	1.000 ± 0.000	0.999 ± 0.000	162.149 ± 199.354
	PNB	0.274 ± 0.437	0.636 ± 0.219	0.268 ± 0.440	1.000 ± 0.000	0.271 ± 0.439	74.266 ± 74.840
	GloVe CVDD	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	259.308 ± 13.955
	BERT CVDD	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	319.108 ± 17.433
	CC	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	15.610 ± 0.055
MedicalTranscriptions	Linear OCSVM	0.971 ± 0.006	0.743 ± 0.007	0.709 ± 0.012	0.496 ± 0.015	0.583 ± 0.012	77.501 ± 1.400
	RBF OCSVM	0.973 ± 0.007	0.734 ± 0.011	0.766 ± 0.010	0.475 ± 0.023	0.586 ± 0.019	94.011 ± 2.294
	Sigmoid OCSVM	0.971 ± 0.006	0.741 ± 0.008	0.707 ± 0.012	0.492 ± 0.016	0.578 ± 0.012	83.649 ± 4.104
	Poly OCSVM	0.973 ± 0.002	0.667 ± 0.002	0.981 ± 0.005	0.335 ± 0.005	0.499 ± 0.005	110.847 ± 6.556
	IsolFor	0.039 ± 0.000	0.500 ± 0.000	0.039 ± 0.000	$\mathbf{1.000 \pm 0.000}$	0.076 ± 0.000	274.933 ± 3.596
	OCRF	0.996 ± 0.000	$\mathbf{0.997 \pm 0.000}$	0.958 ± 0.000	1.000 ± 0.000	0.977 ± 0.000	160.931 ± 196.420
	PNB	0.354 ± 0.227	0.635 ± 0.139	0.486 ± 0.419	0.845 ± 0.189	0.454 ± 0.252	81.234 ± 50.569
	GloVe CVDD	0.989 ± 0.000	0.914 ± 0.002	$\mathbf{0.992 \pm 0.000}$	0.997 ± 0.000	$\mathbf{0.994 \pm 0.000}$	267.377 ± 27.111
	BERT CVDD	0.974 ± 0.124	0.906 ± 0.259	0.986 ± 0.192	0.974 ± 0.175	0.982 ± 0.184	307.439 ± 37.084
	CC	$\mathbf{0.997 \pm 0.008}$	0.989 ± 0.004	0.970 ± 0.017	0.980 ± 0.096	0.973 ± 0.001	$\mathbf{11.435 \pm 0.062}$
MoviePlots	Linear OCSVM	0.635 ± 0.035	0.595 ± 0.026	0.489 ± 0.054	0.470 ± 0.215	0.440 ± 0.106	664.580 ± 4.535
	RBF OCSVM	0.596 ± 0.078	0.589 ± 0.022	0.465 ± 0.061	0.565 ± 0.276	0.458 ± 0.097	718.690 ± 5.346
	Sigmoid OCSVM	0.658 ± 0.019	0.570 ± 0.031	0.530 ± 0.049	0.294 ± 0.184	0.336 ± 0.127	699.067 ± 11.559
	Poly OCSVM	0.617 ± 0.084	0.584 ± 0.030	0.523 ± 0.098	0.478 ± 0.337	0.402 ± 0.151	624.270 ± 3.389
	IsolFor	0.339 ± 0.000	0.500 ± 0.000	0.339 ± 0.000	$\mathbf{1.000 \pm 0.000}$	0.507 ± 0.000	262.591 ± 13.770
	OCRF	0.851 ± 0.000	0.895 ± 0.000	0.660 ± 0.000	0.922 ± 0.003	0.795 ± 0.000	161.427 ± 197.625
	PNB	0.875 ± 0.143	0.910 ± 0.068	$\mathbf{0.972 \pm 0.022}$	0.885 ± 0.156	0.917 ± 0.099	281.049 ± 22.044
	GloVe CVDD	0.822 ± 0.020	0.801 ± 0.020	0.799 ± 0.018	0.928 ± 0.020	0.859 ± 0.017	274.335 ± 56.185
	BERT CVDD	0.817 ± 0.020	0.789 ± 0.020	0.776 ± 0.018	0.905 ± 0.020	0.843 ± 0.017	314.118 ± 46.387
	CC	$\mathbf{0.953 \pm 0.004}$	$\mathbf{0.947 \pm 0.003}$	0.934 ± 0.011	0.931 ± 0.008	$\mathbf{0.931 \pm 0.005}$	$\mathbf{17.512 \pm 0.028}$
Wikileaks	Linear OCSVM	0.999 ± 0.008	0.717 ± 0.058	1.000 ± 0.000	0.434 ± 0.117	0.596 ± 0.117	9.993 ± 0.152
	RBF OCSVM	0.999 ± 0.005	0.708 ± 0.037	1.000 ± 0.000	0.416 ± 0.074	0.584 ± 0.072	11.337 ± 0.272
	Sigmoid OCSVM	0.999 ± 0.003	0.745 ± 0.024	1.000 ± 0.000	0.491 ± 0.040	0.658 ± 0.037	10.628 ± 0.356
	Poly OCSVM	0.999 ± 0.002	0.680 ± 0.019	1.000 ± 0.000	0.361 ± 0.039	0.529 ± 0.041	$\mathbf{9.783 \pm 0.167}$
	IsolFor	0.999 ± 0.008	0.833 ± 0.058	1.000 ± 0.000	0.667 ± 0.117	0.794 ± 0.081	287.498 ± 1.478
	OCRF	0.999 ± 0.000	1.000 ± 0.000	0.999 ± 0.000	1.000 ± 0.000	0.999 ± 0.000	164.122 ± 201.317
	PNB	0.087 ± 0.244	0.541 ± 0.122	0.070 ± 0.248	1.000 ± 0.000	0.074 ± 0.247	78.363 ± 49.268
	GloVe CVDD	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	253.909 ± 28.636
	BERT CVDD	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	298.142 ± 38.494
	CC	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	11.163 ± 0.014

accuracy, precision, recall, F1 score, and time 20 times per dataset. We report mean and standard deviation values.

In order to be able to compare compute times, all models will be run on the same free instance of Google Colaboratory (Google, 2019). Our evaluation instance had a single core running at 2.00GHz, and had access to 13 Gb of RAM.

Finally, we discuss the various VSM models used, comparing the baseline VSMs with NE-TF.

6 Results

Performance metrics can be found in Table 1.

6.1 Predictive Power

CC outperforms baseline models in most scenarios, being the only model with mean accuracies consistently above 95%, balanced accuracies above 94%, and precision, recall, and F1 scores above 93%. PNB had the largest variability out of all algorithms both on a dataset level as well as on a per run level, showcasing how dependent it is on the exact distribution of words that exist within the unlabeled set. OCRF is one of the best performing baseline models, while IsolFor performed the worst, clearly showing that the splitting algorithm used to determine tree structure is crucial for topic determination with ensemble models. The Linear OCSVM outperformed OCSVM alternatives.

The performance delta between BERT and GloVe does not justify the additional computation costs involved with using a BERT encoding for our problem space. Both neural NLP models are consistently outperformed by CC across datasets for one class topic determination. While CVDD and other neural NLP algorithms that use embeddings have use cases in one-class topic determination where they work well, they perform worse when the positive class is highly manifold in nature as is the case for the Jobs and MoviePlots datasets.

6.2 Computation Efficiency

Where CC truly shines is in computational efficiency, showcased in the scenarios with high text complexity. Since we compare each evaluation vector to the max and min vectors, CC has a worse case runtime efficiency of O(dn), where d is the vector dimension number and n is the number of vectors to be evaluated. In practice however, the efficiency is much greater, as we can short-circuit computation as soon as we find a discrepancy; this is a benefit that none of the baselines have. When we compare this to ensembles with a runtime of O(d*nlog(n)), kernel OCSVMs with a runtime of O($n_{support}$*dn) where $n_{support}$ is the number of support vectors, PNB which has a runtime of O(dn + 4d) due to performing training and evaluation at the same time, and neural NLP solutions having a forward pass complexity of $O(n^4)$ (Fredenslund, 2018), the efficiency of CC is clear.

Linear OCSVM has the highest computation efficiency out of the baselines, with a similar worse case runtime efficiency as CC of O(dn). However for each vector at each dimension, Linear OCSVM performs two operations compared to only one, a multiplication as well as an addition. Additionally, Linear OCSVM has no short-circuit capability, so it will always take the maximal amount of time to compute. This can be seen in our results, where CC outperforms Linear OCSVM in computation time especially on the more complex datasets like MedicalTranscriptions and MoviePlots where the time differences are stark.

6.3 VSM Comparison

We identified that the encoding and embedding process is the foremost reason behind the long computation times both versions of CVDD has. This is the reason behind the development of NE-TF; being a bag-of-words VSM it boasts great speed in creating its vector representations. Additionally, bag-of-words VSM models like NE-TF also provides benefits in terms of memory footprint; for our datasets, SpacyEncoding requires 154.7MB, BertTokenizer requires 157.1MB, while NE-TF requires only 18.3MB leading to a roughly 9 times smaller footprint.

When we compare to alternative bag-of-words VSM models NE-TF has a comparable memory footprint but is faster to compute; the statistical significance weighting mitigates the need for stop word pruning, further improving performance.

7 Conclusion

We show that Conical Classification is a computationally efficient method of one-class topic classification that aims to identify whether a vector is within the conical span of the training corpus for a given topic. When combined with Normal Exclusion, Conical Classification showcases the optimal combination of predictive power, consistently great results, and fast computation times.

References

Abien Fred Agarap and Paul Grafilon. 2018. Statistical analysis on e-commerce reviews, with sentiment classification using bidirectional recurrent neural network (rnn). *arXiv preprint arXiv:1805.03687*.

Hadeer Ahmed, Issa Traore, and Sherif Saad. 2017. Detection of online fake news using n-gram analysis and machine learning techniques. In *International conference on intelligent, secure, and dependable systems in distributed and cloud environments*, pages 127–138. Springer.

Hadeer Ahmed, Issa Traore, and Sherif Saad. 2018. Detecting opinion spams and fake news using text classification. *Security and Privacy*, 1(1):e9.

Bandar Alghamdi, Fahad Alharby, et al. 2019. An intelligent model for online recruitment fraud detection. *Journal of Information Security*, 10(03):155.

Maik Anderka, Benno Stein, and Nedim Lipka. 2011. Detection of text quality flaws as a one-class classification problem. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2313–2316.

Yeshwanth Balachander and Teng-Sheng Moh. 2018. Ontology based similarity for information technology skills. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 302–305. IEEE.

John H Beattie and Mark P Richards. 1994. Separation of metallothionein isoforms by micellar electrokinetic capillary chromatography. *Journal of Chromatography A*, 664(1):129–134.

Thorsten Brants and Alex Franz. 2006. 1t 5-gram version 1.0. *Linguistic Data Consortium*.

Rhodessa J Cascaro, Bobby D Gerardo, and Ruji P Medina. 2019. Aggregating filter feature selection methods to enhance multiclass text classification. In *Proceedings of the 2019 7th International Conference on Information Technology: IoT and Smart City*, pages 80–84.

Pratik Chattopadhyay, Lipo Wang, and Yap-Peng Tan. 2018. Scenario-based insider threat detection from cyber activities. *IEEE Transactions on Computational Social Systems*, 5(3):660–675.

Chandler Davis. 1954. Theory of positive linear dependence. *American Journal of Mathematics*, 76(4):733–746.

Francois Denis, Anne Laurent, Rémi Gilleron, and Marc Tommasi. 2003. Text classification and co-training from positive and unlabeled examples. In *Proceedings of the ICML 2003 workshop: the continuum from labeled to unlabeled data*, pages 80–87.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Giacomo Domeniconi, Gianluca Moro, Roberto Pasolini, and Claudio Sartori. 2015. A comparison of term weighting schemes for text classification and sentiment analysis with a supervised variant of tf. idf. In *International Conference on Data Management Technologies and Applications*, pages 39–58. Springer.

George Forman. 2008. Bns feature scaling: an improved representation over tf-idf for svm text classification. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 263–270.

George Forman et al. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3(Mar):1289–1305.

K Fredenslund. 2018. Computational complexity of neural networks.

Joshua Glasser and Brian Lindauer. 2013. Bridging the gap: A pragmatic approach to generating insider threat data. In *2013 IEEE Security and Privacy Workshops*, pages 98–104. IEEE.

Nicolas Goix, Nicolas Drougard, Romain Brault, and Mael Chiapino. 2017. One class splitting criteria for random forests. In *Asian Conference on Machine Learning*, pages 343–358. PMLR.

Google. 2019. Google colaboratory.

Ari Aulia Hakim, Alva Erwin, Kho I Eng, Maulahikmah Galinium, and Wahyu Muliady. 2014. Automated document classification for news article in bahasa indonesia based on term frequency inverse document frequency (tf-idf) approach. In *2014 6th international conference on information technology and electrical engineering (ICITEE)*, pages 1–4. IEEE.

Kathryn Hempstalk, Eibe Frank, and Ian H Witten. 2008. One-class classification by combining density and class probability estimation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 505–519. Springer.

Chenlong Hu, Yukun Feng, Hidetaka Kamigaito, Hiroya Takamura, and Manabu Okumura. 2021. One-class text classification with multi-modal deep support vector data description. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3378–3390, Online. Association for Computational Linguistics.

Erel Joffe, Emily J Pettigrew, Jorge R Herskovic, Charles F Bearden, and Elmer V Bernstam. 2015. Expert guided natural language processing using one-class classification. *Journal of the American Medical Informatics Association*, 22(5):962–966.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Donghwa Kim, Deokseong Seo, Suhyoun Cho, and Pilsung Kang. 2019a. Multi-co-training for document classification using various document representations: Tf–idf, lda, and doc2vec. *Information Sciences*, 477:15–29.

Jeongrae Kim, Han-Joon Kim, and Hyoungrae Kim. 2019b. Fraud detection for job placement using hierarchical clusters-based deep neural networks. *Applied Intelligence*, 49(8):2842–2861.

Sang-Woon Kim and Joon-Min Gil. 2019. Research paper classification systems based on tf-idf and lda schemes. *Human-centric Computing and Information Sciences*, 9(1):1–21.

Theodora Kousta and Clement S Bellet. Local interpretable model-agnostic explanations for long short-term memory network used for classification of amazon customer reviews.

Duc C Le, Sara Khanchi, A Nur Zincir-Heywood, and Malcolm I Heywood. 2018. Benchmarking evolutionary computation approaches to insider threat detection. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1286–1293.

Duc C Le and A Nur Zincir-Heywood. 2019. Machine learning based insider threat modelling and detection. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 1–6. IEEE.

Thanh-Dung Le, Rita Noumeir, Jerome Rambaud, Guillaume Sans, and Philippe Jouvet. 2021. Machine learning based on natural language processing to detect cardiac failure in clinical narratives. *arXiv preprint arXiv:2104.03934*.

Kathy Lee, Diana Palsetia, Ramanathan Narayanan, Md Mostofa Ali Patwary, Ankit Agrawal, and Alok Choudhary. 2011. Twitter trending topic classification. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 251–258. IEEE.

Xiaoxin Lin. 2020. Sentiment analysis of e-commerce customer reviews based on natural language processing. In *Proceedings of the 2020 2nd International Conference on Big Data and Artificial Intelligence*, pages 32–36.

Cai-zhi Liu, Yan-xiu Sheng, Zhi-qiang Wei, and Yong-Quan Yang. 2018. Research of text classification based on improved tf-idf algorithm. In *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, pages 218–222. IEEE.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE.

Syed Mahbub and Eric Pardede. 2018. Using contextual features for online recruitment fraud detection.

Larry M Manevitz and Malik Yousef. 2001. One-class svms for document classification. *Journal of machine Learning research*, 2(Dec):139–154.

Justin Martineau and Tim Finin. 2009. Delta tfidf: An improved feature space for sentiment analysis. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 3.

Fanzhi Meng, Fang Lou, Yunsheng Fu, and Zhihong Tian. 2018. Deep learning based attribute classification insider threat detection for data security. In *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pages 576–581. IEEE.

Francesco Moramarco, Damir Juric, Aleksandar Savkov, and Ehud Reiter. 2021. Towards objectively evaluating the quality of generated medical summaries. *arXiv preprint arXiv:2104.04412*.

MTSamples. Collection of transcribed medical transcription sample reports and examples.

Oxford. 2011. The oec: Facts about the language. *Oxford English Dictionary*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

PromptCloud. 2017. Us jobs kaggle dataset.

CR Rao and Venkat N Gudivada. 2018. *Computational analysis and understanding of natural languages: Principles, methods and applications*. Elsevier.

M Niharika Reddy, T Mamatha, and A Balaram. 2018. Analysis of e-recruitment systems and detecting e-recruitment fraud. In *International Conference on Communications and Cyber Physical Engineering 2018*, pages 411–417. Springer.

Lukas Ruff, Yury Zemlyanskiy, Robert Vandermeulen, Thomas Schnake, and Marius Kloft. 2019. Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4061–4071.

Atul Sajjanhar, Yong Xiang, et al. 2019. Image-based feature representation for insider threat classification. *arXiv preprint arXiv:1911.05879*.

Kwang-Kyu Seo. 2007. An application of one-class support vector machines in content-based image retrieval. *Expert Systems with Applications*, 33(2):491–498.

Pinky Sitikhu, Kritish Pahi, Pujan Thapa, and Subarna Shakya. 2019. A comparison of semantic similarity methods for maximum human interpretability. In *2019 artificial intelligence for transforming business and society (AITB)*, volume 1, pages 1–4. IEEE.

Frank Stappen. 2020. Motion and manipulation lecture series. *Utrecht University*.

Hao Sun, Tao Jiang, and Yugang Dai. 2019. Sentiment analysis of commodity reviews based on multilayer lstm network. In *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, pages 1–5.

Rachael Tatman. 2017. $\frac{1}{3}$ million most frequent english words on the web. *Kaggle*.

Bruno Trstenjak, Sasa Mikac, and Dzenana Donko. 2014. Knn with tf-idf based framework for text categorization. *Procedia Engineering*, 69:1356–1364.

Aaron Tuor, Samuel Kaplan, Brian Hutchinson, Nicole Nichols, and Sean Robinson. 2017. Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. *arXiv preprint arXiv:1710.00811*.

Sokratis Vidros, Constantinos Kolias, Georgios Kambourakis, and Leman Akoglu. 2017. Automatic detection of online recruitment frauds: Characteristics, methods, and a public dataset. *Future Internet*, 9(1):6.

Sida I Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94.

Yichen Wei, Kam-Pui P Chow, and Siu-Ming Yiu. 2021. Insider threat prediction based on unsupervised anomaly detection scheme for proactive forensic investigation. *Digital Investigation*.

Shuwei Xiao and Weiqin Tong. 2021. Prediction of user consumption behavior data based on the combined model of TF-IDF and logistic regression. *Journal of Physics: Conference Series*, 1757(1):012089.

Wen Zhang, Taketoshi Yoshida, and Xijin Tang. 2011. A comparative study of tf* idf, lsi and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765.

Ling Zhuang and Honghua Dai. 2006. Parameter estimation of one-class svm on imbalance text classification. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 538–549. Springer.

Guido Zuccon, Daniel Kotzur, Anthony Nguyen, and Anton Bergheim. 2014. De-identification of health records using anonym: Effectiveness and robustness across datasets. *Artificial intelligence in medicine*, 61(3):145–151.