# ONE CLASS CONVOLUTIONAL NEURAL NETWORK (OC-CNN): NOVELTY DETECTION IN MISINFORMATION TWEETS

*Rachel Rolle*

American University

## ABSTRACT

The purpose of this research is to propose a method on effectively detecting misinformation shared on social media. Using One Class Convolutional Neural Network (OC-CNN), the hope is to detect these posts regardless of volume and ratio of reliable and unreliable posts. Others have run experiments prior, but I will be using an imbalance set of data to mimic a real case scenario. More specifically, I will be using tweets on the topic of COVID-19.

***Index Terms—*** one class cnn, social media, twitter, deep learning, novelty detection, imbalance dataset, COVID-19

## 1. INTRODUCTION

In this digital era, social media platforms are utilized by millions worldwide, facilitating the rapid dissemination of information with minimal censorship due to the principle of freedom of expression [1]. Consequently, the proliferation of misinformation, whether intentional or inadvertent, poses significant challenges, potentially influencing individuals' behaviors negatively [2]. Detecting such misinformation swiftly on social media platforms is crucial. However, the task is hindered by the sheer volume of data, ambiguous content, diverse post structures, and imbalanced class distributions (true and false tweets).

Despite its inherent difficulties, addressing this challenge is imperative, particularly evident during events like the COVID-19 pandemic. The rampant spread of misinformation during this period significantly impacted public perception, contributing to heightened anxiety levels and irrational behaviors, such as panic buying triggered by false claims of shortages [3]. Such instances underscore the urgency of combating malicious content dissemination on social media platforms.

To address this issue, this project aims to leverage an active learning approach known as one-class convolutional neural network (OC-CNN) to identify posts containing misinformation. Successful implementation of this approach would enable rapid identification of false tweets.


**Figure 1: Distribution in Training Set**

Novelty detection, a core concept in this context, involves training the model to identify the class that deviates from the norm [4]. In our case, this entails introducing a labeled dataset comprising false tweets to the model, which then learns to differentiate between reliable and unreliable tweets. However, there's a risk of the model inadvertently learning features of both reliable and unreliable tweets, undermining its ability to accurately identify false tweets.

One potential challenge with one-class novelty detection is the risk of overfitting, where the model becomes overly specialized on the training data, limiting its generalization to new data, such as the test set. Additionally, the complexity of Twitter posts, which may include multimedia elements like videos, emojis, and images, presents another challenge, as the model may not adequately capture all relevant factors contributing to false labeling. In order to isolate the imbalance problem, text tweets will only be presented to the model.

## 2. RELATED WORK

The 2021 publication titled "One-Class Classification: A Survey" conducted a thorough exploration of one-class

classification methodologies. This survey meticulously examined a range of techniques, shedding light on their individual merits and constraints [4]. While the discussion encompassed novelty detection, the primary focus remained on image detection rather than text classification.
This observation sparked my interest in adapting the principles of one-class convolutional neural network (CNN) models for text classification.

Considering the prevalent challenge of imbalanced data distribution between reliable and unreliable posts, I propose a novel approach. This strategy involves training the model exclusively on features associated with the underrepresented class—unreliable posts [1]. By adopting this targeted approach, I anticipate a significant enhancement in performance, as the model can allocate its resources solely to discerning characteristics relevant to the minority class.

## 3. METHODOLOGY

In addition to using One-Class CNN to train our model, the following methods were used in this project: imbalance classes, text-cleaning, Word2Vec, One-class novelty detection, and Principal Component Analysis (PCA).

### 3.1. Data Acquisition

The dataset utilized in this study was sourced from Dr. Zois Boukouvalas, who conducted a Neural Network/ Deep Learning course. This dataset comprised tweets extracted from Twitter pertaining to the COVID-19 event. Each tweet was labeled as either "reliable" (0) or "unreliable" (1). Additionally, a separate corpus exclusively containing false posts from various social media platforms was obtained. For the purpose of this study, the dataset containing only false tweets is referred to as the "train set," while the dataset containing both false and true tweets is referred to as the "test set."

### 3.2. Imbalance Classes

Initially, the test set obtained from Dr. Boukouvalas was balanced, consisting of 280 false and 280 true tweets. However, to simulate real-world scenarios where false information is significantly outnumbered by true information, a deliberate imbalance in class distribution was created. To achieve this, a Python function was developed to randomly remove 250 tweets from the false class. A fixed random state (42) was set to ensure consistency across multiple runs of the code.

As shown in Figure 2, the false tweets are greatly reduced and is approximately 10% of the test set. Now that our imbalance set is prepared, we can continue on to the next stage.

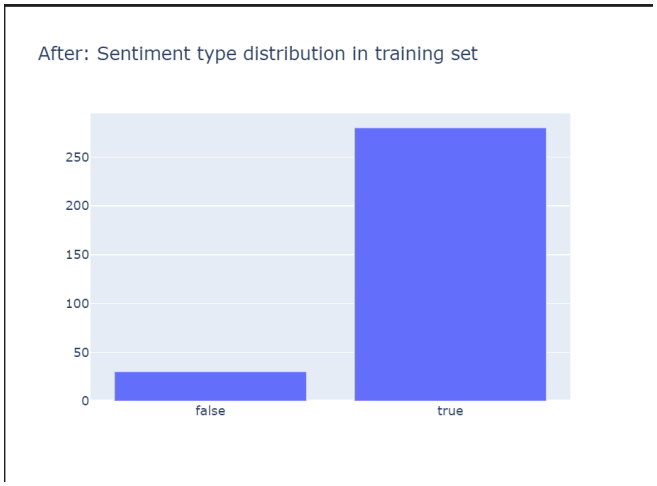


**Figure 2: Create Imbalance in False Tweets**

### 3.3. Text Preprocessing

We're working with textual data, so it's crucial to standardize each feature. This involves several steps: converting all text to lowercase, stemming words to their base form, and eliminating common stop words. These actions help minimize irrelevant parts of the corpus and enhance overall results. For instance, frequently occurring stop words like "and," "or," and "so" are ubiquitous but unlikely to differentiate between false and true tweets, so it's sensible to exclude them from evaluation altogether.

In our text cleaning function, we not only remove stop words but also eliminate hashtags, links, and emojis. While these elements may carry some significance, they're often too unique to establish meaningful connections within each post. For instance, the same link is unlikely to be shared across multiple posts, so its presence doesn't add value to the analysis. While researchers may opt to retain emojis and hashtags depending on the project's scope and timeline, for this assignment, we've chosen to disregard them. The removal of these features is facilitated using the "re" package.

Another essential text normalization technique is the "WordNetLemmatizer" function from the "nltk.stem" package. This function reduces words to their root form by removing suffixes, ensuring that different forms of the same word are treated as a single entity.

To apply each operation, the text must first be tokenized, splitting it into individual elements. For example, the sentence:

[“ I went to a birthday party.”]

would be first tokenized and would be represented in a list as:

[ ‘I’, ‘went’, ‘to’, ‘a’, ‘birthday’, ‘party’, ‘.’]

After processing through the "clean_text" function, these individual words are recombined into a coherent sentence, such as:

[‘went birthday party’]

### 3.4. Word2Vec Feature Extraction

The next step in our methodology was to convert the textual data into numerical features that could be processed by our classifier. To accomplish this, we utilized the Word2Vec technique. Word2Vec is a popular word embedding method that represents words as dense vectors in a continuous vector space. These word vectors are trained to capture semantic relationships between words based on their distributional properties in a large corpus of text.

In our study, we employed a pretrained Word2Vec model provided by Google. This model has been trained on a massive dataset comprising billions of words from various sources, such as news articles, web pages, and other textual content available on the internet [5]. By leveraging this pretrained model, we were able to benefit from the rich semantic information it had already learned during training.

Using Word2Vec, each word in our dataset was mapped to a high-dimensional vector representation. These vectors encode semantic information about the words' meanings and relationships with other words in the vocabulary[6]. For example, words that frequently co-occur in similar contexts tend to have similar vector representations, reflecting their semantic similarity.

Once we obtained the word vectors using Word2Vec, we employed cosine similarity as a metric to quantify the similarity between pairs of word vectors. Cosine similarity measures the cosine of the angle between two vectors and provides a measure of their similarity, with values ranging from -1 to 1 [7]. A cosine similarity of 1 indicates that the vectors are identical, while a value of -1 indicates that they are diametrically opposed.

By calculating cosine similarity between word vectors, we were able to quantify the semantic similarity between words in our dataset. This allowed us to capture the underlying semantic structure of the text data and extract meaningful features that could be used by our classifier to distinguish between different classes of tweets (e.g., reliable vs. unreliable).
The cosine similarity is expressed as follows in “Equation 1” [8]:

**Equation 1: Cosine Similarity Equation [8]**

$$similarity = \cos\theta = \frac{\bar{x} \bullet \bar{y}}{\| \bar{x} \| \| \bar{y} \|}$$

where :
$\bar{x} \bullet \bar{y}$ : Vector dot product from x and y. $\sum_{k=1}^{n} x_k y_k$
$\|x\|$ : Long vector $x$. $\sum_{k=1}^{n} x_k^2$
$\|y\|$ : Long vector $y$, $\sum_{k=1}^{n} y_k^2$

### 3.5. One-Class Novelty Detection

The approach of one-class novelty detection shares similarities with One-Class classification [4]. The fundamental concept revolves around delineating a decision boundary around a single class, achieved by training a model exclusively on that class. In our case, we apply this principle to an imbalanced dataset by exposing the one-class convolutional neural network (OC-CNN) solely to false tweets.

Conversely, the positive class—true tweets—remains unseen in the test set, constituting the "novelty" data, representing new and unseen patterns within the data [8]. The objective is for the OC-CNN model to become proficient in identifying false tweets to such an extent that it can effectively detect new tweets introduced during testing. Consequently, this results in the establishment of a decision boundary around the novel class.

This approach is advantageous, especially considering that the false class represents only 10% of the population. Attempting a random test-train-split without providing the model exposure to the true tweets would likely lead to overfitting and hinder the model's ability to detect false tweets. Indeed, in our experiment, this scenario was attempted, but false tweets went undetected. Despite efforts to adjust the detection threshold, as will be discussed in the subsequent section, the model still struggled to detect false tweets.

A commonly suggested solution to address imbalanced datasets is to balance the dataset using techniques such as Under-Sampling or Over-Sampling. However, employing such techniques would undermine the real-world scenario we aim to replicate. Therefore,

maintaining the imbalance in the dataset is crucial for the authenticity of our experimental setup.

### 3.6. One-Class Convolutional Neural Network (OCCNN)

One-class convolutional neural networks (OC-CNN) are a type of neural network specifically designed for anomaly detection and one-class classification tasks. Unlike traditional convolutional neural networks (CNNs) that are trained on both positive and negative examples, OC-CNNs are trained only on instances from the target class, which is often the minority or underrepresented class in a dataset.

The key idea behind OC-CNN is to learn a representation of the target class that is sufficiently distinct from the background or outlier data [9]. This is achieved by training the network to identify patterns and features that are unique to the target class, while disregarding features present in the outlier data.

In the context of solving the imbalance dataset problem of classifying false (underrepresented class) and true tweets, OC-CNN can be utilized as follows [9]:

1. Training on False Tweets Only: OC-CNN will be trained exclusively on a dataset containing false tweets, which represent the minority or underrepresented class. This allows the network to learn representations and features specific to false tweets.
2. Learning Discriminative Features: During training, the OC-CNN learns to extract discriminative features from false tweets that distinguish them from true tweets. These features may include textual patterns, linguistic structures, or contextual information unique to false tweets.
3. Establishing Decision Boundary: By learning these discriminative features, the OC-CNN effectively establishes a decision boundary that separates false tweets from true tweets in the feature space. This decision boundary serves as the basis for classifying unseen tweets during testing.
4. Anomaly Detection: During testing, the trained OC-CNN can classify tweets as either false or true based on their similarity to the learned representations of false tweets. Tweets that fall within the decision boundary are classified as true, while those outside the boundary are classified as false, representing anomalies or outliers.

An equation that encapsulates the essence of OC-CNN for one-class classification tasks is as follows:

$$y\hat{}i=\sigma(f(xi;\theta))$$

Where:

- $y\hat{}i$ represents the predicted label for instance $xi$
- $f(xi;\theta)$ denotes the mapping function learned by the OC-CNN with parameters $\theta$.

$\sigma$ is the activation function (e.g., sigmoid or softmax) that maps the output of $f(xi;\theta)$ to a probability distribution over the classes[9].

### 3.6. Principal Component Analysis (PCA)

To assess the parameters, we employed a visualization technique to examine how the model discriminated between true and false tweets based on their features. Given the high-dimensional nature of the data, Principal Component Analysis (PCA) was utilized to reduce its dimensionality. PCA enables us to condense the data into a simplified, low-dimensional representation, extracting the most salient information [10].

In this study, the word2vec feature extraction resulted in multiple features representing each word. While this facilitates capturing semantic relationships, it complicates interpreting the model's decision-making process. Figure 3 illustrates the application of PCA from the "sklearn.decomposition" package to the text data using the ".fit_transform" method.

```
from sklearn.decomposition import PCA

X = dense_test_X

pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)
```

**Figure 3: Applying PCA to Text Features**

## 4. RESULTS

**Table 1: Model's Classification Report**

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| 0 (True) | 0.90 | 1.00 | 0.95 | 280 |
| 1 (False) | 0.00 | 0.00 | 0.00 | 30 |
| accuracy | | | 0.90 | 310 |
| Macro avg. | 0.45 | 0.50 | 0.47 | 310 |
| Weighted avg | 0.82 | 0.90 | 0.86 | 310 |

Based on the results, the OC-CNN model seems to perform exceptionally well in identifying true tweets (class 0) with high precision, recall, and F1-score. However, it failed to correctly identify any false tweets (class 1) as indicated by zero precision, recall, and F1-score for that class.

This outcome suggests that the model was effective in capturing the characteristics and patterns specific to true tweets, but it struggled to generalize to false tweets, known as overfitting.

## 5. DISCUSSION

Given that this problem performed better with the One-Class Support Vector Machine (OCSVM) it suggests that the OC-CNN model architecture may not be suitable for capturing nuances of false tweets. One-Class CNNs require careful design and tuning to balance model complexity with the complexity of the problem. If the model is too simple or too complex. It may not generalize well to unseen data. Since there was prior observation to a simpler model completing this task, Figure 5, it suggests that the OC-CNN is too complex for a problem like this. To resolve this problem, it may come down to a limitation in resources, for this research project, there were 40,000 labelled false posts available, yet it may need more diverse data to enrich the training dataset and improve the model's understanding of this class.

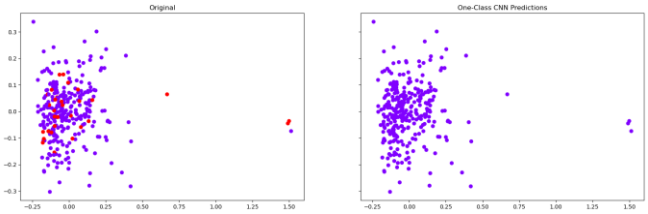


**Figure 4: OC-CNN Classification**

## 6. CONCLUSION

In conclusion, the superior performance of OCSVM compared to OC-CNN suggests that the OC-CNN model architecture may not be optimal for capturing the nuances of false tweets. While OC-CNN offers potential for sophisticated feature extraction and representation learning, its effectiveness heavily relies on its ability to generalize to unseen data. One-Class CNNs require careful design and tuning to balance model complexity with the problem at hand. If the architecture is too simplistic, it may fail to capture intricate patterns, while excessive complexity may lead to overfitting and poor generalization.

The observation of simpler models successfully completing similar tasks raises questions about whether the additional complexity introduced by OC-CNN is necessary or beneficial. Additionally, limitations in achieving satisfactory results with OC-CNN may stem from resource constraints, such as the size and diversity of the training dataset. Despite access to 40,000 labeled posts, lack of diversity within the data may have hindered OC-CNN's ability to learn representations for false tweets. Future research could focus on acquiring more diverse data to enhance model performance.

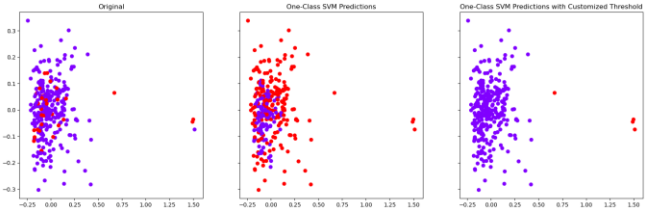


**Figure 5: OCSVM Classification**

## 11. REFERENCES

[1] Boukouvalas, Z., Shafer, A.: Role of statistics in detecting misinformation: A review of the state of the art, open issues, and future research directions. Annual Review of Statistics and Its Application 11 (2024)

[2] Guacho, G.B., Abdali, S., Shah, N., Papalexakis, E.E.: Semi-supervised contentbased detection of misinformation via tensor embeddings. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 322–325 (2018). IEEE

[3] Leung, J., Chung, J.Y.C., Tisdale, C., Chiu, V., Lim, C.C., Chan, G.: Anxiety and panic buying behaviour during covid-19 pandemic—a qualitative analysis of toilet paper hoarding contents on twitter. International journal of environmental research and public health 18(3), 1127 (2021)

[4] Perera, Pramuditha, Poojan Oza, and Vishal M. Patel. "One-class classification: A survey." *arXiv preprint arXiv:2101.03064* (2021).

[5] Google. "Word2Vec." Google Code Archive. Accessed 25 Apr. 2024, https://code.google.com/archive/p/word2vec/.

[6] Mikolov, Tomas, et al. "Efficient Estimation of Word Representations in Vector Space." arXiv preprint arXiv:1301.3781 (2013).

[7] Rehurek, Radim, and Petr Sojka. "Software Framework for Topic Modelling with Large Corpora." Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, 2010, pp. 45-50.

[8] Jatnika, Derry, Moch Arif Bijaksana, and Arie Ardiyanti Suryani. "Word2vec model analysis for semantic similarities in english words." *Procedia Computer Science* 157 (2019): 160-167.

[9] "Convolutional Neural Networks." Neural Networks and Deep Learning, Deep Learning,www.deeplearningbook.org/contents/convnets.html.

[10] Pimentel, Marco AF, et al. "A review of novelty detection." *Signal processing* 99 (2014): 215-249.

[11] S. Ravanbakhsh, et al. "Deep Learning for Anomaly Detection: A Review." arXiv preprint arXiv:2009.11732 (2020).