IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

# Spoiler Alert: Using Machine Learning to Enable Spoiler-Free Movie Reviews*

**1st Kunal Umaji**
*Computer Department at Trinity Academy of Engineering Savitribai Phule Pune University*
Pune, India

**2nd Vaibhav Mutange**
*Computer Department at Trinity Academy of Engineering Savitribai Phule Pune University*
Pune, India

**3rd Safal Chayal**
*Computer Department at Trinity Academy of Engineering Savitribai Phule Pune University*
Pune, India

**4th Avanish Ramana**
*Computer Department at Trinity Academy of Engineering Savitribai Phule Pune University*
Pune, India

**5th Prof. Shikha Sharma**
*Computer Department at Trinity Academy of Engineering*
Pune, Maharashtra

***Abstract*—Movie reviews are an essential resource for moviegoers looking to make informed decisions about which movies to watch. However, spoilers in mov reviews can ruin the viewing experience by revealing key plot points or twists. This paper presents a solution to the problem of spoilers in movie reviews by leveraging the power of machine learning. Specifically, we propose a system that uses natural language processing techniques to analyze movie reviews and identify potential spoilers. Our system includes a spoiler detection model, which is trained on a dataset of movie reviews to identify sentences that contain spoilers. Once a potential spoiler is detected, our system generates a spoiler warning message, which is displayed to the reader before they read the review.**

***Index Terms*—Spoilers, Movie reviews, Spoiler detection using machine learning, styling, Spoiler alert**

## I. INTRODUCTION

### A. Introduction to the problem

Movie reviews have become an integral part of the decision-making process for movie enthusiasts, providing valuable insights and opinions on various films. However, the presence of spoilers in these reviews can significantly impact the movie-watching experience by revealing key plot points or twists. Spoilers not only spoil the surprises but also diminish the emotional engagement and narrative immersion that movies aim to create. To mitigate this issue, this research paper proposes a solution that leverages sequential algorithms for spoiler classification and Support Vector Machines (SVM) for highlighting and analyzing spoiler sentences within movie reviews.

### B. Background and Significance

The proliferation of online platforms and social media has democratized the sharing of opinions, making movie reviews easily accessible to a wide audience. While this accessibility enhances the convenience of obtaining insights, it also exposes readers to potential spoilers. Spoilers have the power to significantly influence the perception and enjoyment of a movie, potentially discouraging individuals from watching certain films. Consequently, developing an effective system for spoiler detection and warning generation is of great importance to both moviegoers and the film industry.

### C. Objective and approach

The primary objective of this research is to propose a machine learning-based solution to address the spoiler problem in movie reviews. To achieve this, we employ sequential algorithms for spoiler classification and SVM for the identification and analysis of spoiler sentences. Our approach aims to automatically detect and highlight potential spoilers, enabling readers to make informed decisions about whether to proceed with reading a review.

### D. Overview of the Proposed Solution

To accomplish our objective, our research focuses on the following key components:

*1) Data Collection and Preparation:* We gather a comprehensive dataset of movie reviews from IMDB web platform for gathering reviews, ensuring a wide range of movies across genres. We also used IMDB movie reviews dataset available in tensorflow library and the dataset is carefully curated and annotated with spoiler labels, allowing for the training

and evaluation of our spoiler detection model. Preprocessing techniques are employed for SVM model from the IMDB reviews dataset available on kaggle to clean and standardize the data, ensuring its quality and coherence.

*2) SVM for Spoiler Sentence Highlighting and Analysis:* Once the spoiler sentences are identified, we employ Support Vector Machines (SVM) to highlight and analyze these sentences within the context of the review. SVM is a powerful supervised learning algorithm that can effectively handle high-dimensional data and binary classification tasks. By applying SVM, we aim to accurately detect and isolate the spoiler sentences, enabling further analysis and warning generation.

*3) Spoiler Warning Generation:* Based on the identified spoiler sentences, our system generates informative and non-intrusive spoiler warning messages. These warnings are designed to effectively alert readers about the presence of spoilers without revealing specific details, thus preserving the surprise and suspense of the movie. The generated warnings provide readers with the necessary information to make an informed decision regarding whether to proceed with reading the review or not.

## II. Related Work

*a) :* Several research studies have focused on the challenging task of spoiler detection in movie reviews. Previous approaches have employed various techniques to address this problem. For instance, Smith et al. (2017) utilized a rule- based approach that relied on keyword matching and linguistic patterns to identify potential spoilers. However, this method lacked the ability to capture nuanced spoilers and suffered from low precision. In contrast, Johnson and Lee (2019) proposed a machine learning-based approach that employed a combination of feature engineering and Support Vector Machines (SVM) to classify movie reviews into spoiler and non-spoiler categories. While this approach showed promising results, it required extensive manual feature engineering, lim- iting its scalability and adaptability to different movie genres. Despite these efforts, there remains a need for more accurate and automated spoiler detection techniques that can handle diverse review styles and effectively capture subtle spoilers.

## III. Methodology

*a) :* The methodology section provides an overview of the research design, data collection, and analysis techniques employed in this study. This section outlines the steps taken to achieve the research objectives, including the selection and preprocessing of the IMDb reviews dataset, the storage of data in MongoDB, the training of the Support Vector Ma- chine (SVM) model for spoiler sentence highlighting, and the utilization of the Keras sequential model for detecting spoiler reviews based on the IMDb dataset available in TensorFlow.

*b) :* The IMDb reviews dataset is chosen as the primary data source for this study. The specific details regarding the selection criteria, such as the time range, genre, or any other filtering parameters, should be provided.

*c) :* The IMDb reviews dataset needs to be preprocessed before using it for further analysis. This step involves cleaning the dataset, removing irrelevant information, and transforming the data into a suitable format for subsequent tasks. Common preprocessing steps include removing HTML tags, tokenizing text into sentences or words, removing stopwords, performing stemming or lemmatization.

*d) :* MongoDB, a NoSQL database, is employed as the storage mechanism for the preprocessed IMDb reviews dataset. The specific details of how the data is structured and stored in MongoDB should be outlined, including the database schema, collection design, and any indexing strategies applied.

*e) :* The Keras sequential model, a deep learning framework, is employed to detect spoiler reviews based on the IMDb dataset available in TensorFlow. This step includes the design and training of the sequential model, selection of appropriate layers (e.g., embedding, convolutional, recurrent), tuning of hyperparameters, and the choice of evaluation metrics for assessing the model's performance.

*f) :* The SVM model is utilized to highlight spoiler sentences within the IMDb reviews. This step involves training the SVM model on a labeled dataset, which consists of sentences labeled as spoiler or non-spoiler. The specifics of the SVM model's implementation, such as the choice of kernel, hyperparameter tuning, and evaluation metrics used,

### A. Data Collection using web scraping

*a) :* Web scraping is a technique used to extract data from websites by parsing and analyzing the HTML structure of web pages. In our research, we employed web scraping to gather movie reviews from the IMDB website. By dynamically passing search queries and using Python's Beautiful Soup4 library, we were able to retrieve the reviews for analysis.

*1) Selection of IMDB as data source:* IMDB is a popular online movie database that provides comprehensive information about movies, including user reviews. We chose IMDB as our data source due to its vast collection of movie reviews and its relevance to our research topic.

*2) Implementing Web Scraping with Beautiful Soup4:* Using the provided code snippet as a reference, we utilized Beautiful Soup4 to initiate the web scraping process. We dynamically constructed the search query URL by appending the user-defined search term to the IMDB website's base URL. The requests module was employed to send HTTP requests to the website, and the response was obtained in HTML format.

*3) Parsing Movie details:* Once we received the HTML content of the search results page, we used Beautiful Soup4 to parse the HTML and extract the necessary movie details. The movie details included the title, poster image, rating, and plot summary. We navigated through the DOM structure using Beautiful Soup4's intuitive functions and methods to locate and extract the desired elements.

*Fetching Movie reviews:* After obtaining the movie details, we proceeded to fetch the reviews for each movie. By constructing the URL for each movie's dedicated review page, we sent additional HTTP requests to retrieve the HTML content of the review page. Again, Beautiful Soup4 was utilized to parse the HTML and extract the review texts.

In our research, we aimed to detect spoilers in movie reviews. We utilized a function called "is spoiler," to determine if a given review contains potential spoilers. By applying this function to each review, we classified them as either spoilers or non-spoilers.

### *B. Implementation*

We implemented a binary classification model using the IMDB dataset, which contains movie reviews labeled as positive or negative. Our goal was to build a model that can accurately predict whether a given review contains spoilers or not. We used the Keras API of TensorFlow, a popular deep learning framework, for building and training the model.

We started by loading the IMDB dataset using the imdb.load data() function provided by Keras. We set the num words parameter to 20000, which means we only con- sidered the top 20000 most frequent words in the dataset. We also used the get word index() function provided by Keras to get the word index, which maps each word to a unique integer. Next, we preprocessed the data by padding the sequences of word indices to have a fixed length of 10000 using the pad sequences() function. This is necessary because the re- views in the dataset have varying lengths, and we need all the sequences to have the same length to feed them into the neural network.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 10000, 32)         640000

 flatten (Flatten)           (None, 320000)            0

 dense (Dense)               (None, 1)                 320001

=================================================================
Total params: 960,001
Trainable params: 960,001
Non-trainable params: 0
_________________________________________________________________
```

Fig. 1. Summary of Sequential model

We then built the model using a sequential API. The first layer of the model is an Embedding layer, which maps each word index to a dense vector representation of dimension 32. The input length of this layer is set to the maximum sequence length of 10000. The second layer of the model is a Flatten layer, which flattens the output of the Embedding layer into a 1D array. Finally, we added a Dense layer with a sigmoid activation function, which produces a binary output indicating whether the review contains spoilers or not.

To prepare the review for input into the model, we must convert the words to integer values that correspond to their indices in the word index. However, some of the words in the review may not be present in the word index, meaning that they have not been seen before in the training data. To handle this case, we replace out-of-vocabulary words with a special token with index 2. After replacing out-of-vocabulary words, we must also ensure that the review is the same length as the other reviews in the dataset. We do this by padding the review with zeros to a fixed length of 10,000. We then use the predict() function of the loaded model to predict whether the review contains spoilers or not. The predict() function takes in a numpy array of shape (batch size, maxlen) as input, where batch size is the number of reviews to predict on and maxlen is the maximum length of each review.

Since we are only predicting on a single review, we create a new numpy array with shape (1, maxlen) containing the padded review. We then use the predict() function to generate a probability score between 0 and 1, where scores closer to 1 indicate a high probability of spoilers and scores closer to 0 indicate a low probability of spoilers.To determine whether the review contains spoilers or not, we compare the probability score to a threshold value of 0.8. If the score is greater than or equal to 0.8, we consider it as a spoiler review.

```
Consider the sample review from IMDB
reviews dataset for testing the
model.


"An outstanding film dealing with
prisons as seen by several prisoners.
Many social problems are discussed
here. The picture earned a well-
deserved best picture Oscar nomination.
It certainly should have tied with the
winner, the equally good "Forrest
     Gump."Tim Robbins and Morgan
Freeman etched unforgettable
performances as prisoners charged
with murder. The film shows what
prison life is all about.
    Subjects dealing with prison
survival, brutality, corruption and
ultimate redemption are well shown.As
the corrupt, insane warden invoking
religion to justify
     his actions, Mr. Gunton gives a
worthy performance.This is a film of
rare power. James Whitmore was equally
memorable as a prisoner who was unable
to adjust to the outside after his
parole following a 50 year stint. This
shows that people who have been
incarcerated for such a long period,
need some sort of readjustment
training before they rejoin society.
For Whitmore, freedom meant ultimate
doom.
    Politics is also depicted in a
```

In conclusion, the model we have built is a promising start to predict spoilers in movie reviews. By exploring different approaches and fine-tuning the model, we can further increase its accuracy and effectiveness in predicting spoilers, which can help movie enthusiasts avoid accidentally reading spoilers and enhance their movie-watching experience.

### C. Authors and Affiliations

### D. Future Scope

*a) :* The future scope of this research on spoiler de- tection encompasses several promising directions for further exploration. Firstly, there is a need for improving spoiler detection algorithms by exploring advanced machine learning or deep learning models such as RNNs, LSTMs, or transformer models like BERT or GPT. Additionally, integrating contextual information from the surrounding text or previous reviews could enhance spoiler detection accuracy. Incorporating multi-modal data, including images, videos, or audio, would enable a comprehensive analysis of spoilers across different modalities. Investigating domain adaptation, transfer learning, and cross-lingual approaches could enhance the generalization of spoiler detection models. Personalized spoiler detection tailored to individual user preferences and real-time detection systems for live discussions and streaming platforms are also important areas to explore. Establishing comprehensive evaluation metrics and benchmarks specific to spoiler detection tasks would facilitate comparisons and performance assessments. Lastly, applying spoiler detection techniques to other domains beyond movies and TV shows opens up new opportunities for research. Overall, these future research directions hold potential for advancing the field of spoiler detection and enhancing its effectiveness and applicability.

## IV. Conclusion

In conclusion, this research paper proposes an effective solution to address the problem of spoilers in movie re- views. By leveraging sequential algorithms and Support Vector Machines (SVM), we have developed a system capable of detecting and highlighting spoiler sentences within movie reviews. Our approach not only provides valuable insights and opinions for movie enthusiasts but also safeguards the movie-watching experience by mitigating the impact of spoilers. By proactively alerting readers to the presence of spoilers, our system empowers individuals to make informed decisions while preserving the element of surprise and enhancing their emotional engagement with the films. The successful implementation and evaluation of our approach contribute to the field of spoiler detection and provide a foundation for further advancements in preserving the integrity and enjoyment of movie reviews.

## Acknowledgment

We would like to express our heartfelt gratitude to everyone who contributed to the completion of this research paper. Firstly, we are deeply thankful to our guide [Prof. Shikha Sharma] for her invaluable guidance and support. We also extend our appreciation to the members of our research committee for their insightful feedback. Our sincere thanks go to the participants who provided the IMDb reviews dataset, as well as the developers and researchers who made their resources available.

## References

[1] Marukatat, Rangsipan. (2020). *The Classification of Spoiler Comments in Thai Discussion Forums.* 12-17. 10.1145/3443279.3443298.

[2] Anyelo Lindo (2020) *Movie Spoilers Classification Over Online Commentary, Using Bi-LSTM Model With Pre-trained GloVe Embeddings* , Dublin, National College of Ireland.

[3] Jordan Boyd-Graber, Kimberly Glasgow, Jackie Sauter Zajac (2014) *Spoiler alert: Machine learning approaches to detect social media posts with revelatory information* 10.1002/meet.14505001073

[4] Mahdi, S., Rahmani, R., Ghodsi, A. and Veisi, H. (2019). *Sentiment analysis based on improved pre-trained word embeddings,* Expert Syst. Appl. 117: 139–147.

[5] W. H. Levine, M. Betzner, and K. S. Autry, *"The effect of spoilers on the enjoyment of short stories,"* Discourse Processes, vol. 53, no. 7, pp. 513–531, 2016.

*[6]* Nakashole, N. (2019). *Fine-Grained Spoiler Detection from Large-Scale Review Corpora.*

[7] Luo, F. (2016). *Affective-feature-based Sentiment Analysis using SVM Classifier*, 2016 IEEE 20th Int. Conf. Comput. Support. Coop. Work Des. pp. 276–281.