CSE343 Machine Learning
Project Interim Report

Authors:
Aniansh Raj Singh (2019019)
Bhavya Narang (2019462)
Harshal Dev(2019306)
Satwik Tiwari (2019100)

Motivation:

- During the pandemic, “Virtual Movie Nights” saw quite a boom.
- However deciding on what to watch, given everyone has different preferences and tastes and liking was a tedious process. This also resulted in having a lot of time and energy being wasted.
- We hope to help **simplify this decision-making process** through our work by building a **movie recommendation system** that takes into account the preferences of all the participants to recommend movies that are aligned with most of their preferences.

Introduction:

Given a certain number of movies by consensus with a group of friends, we are trying to predict a movie that these friends can watch and the predicted movie is going to try to accommodate multiple choices and interests of different individuals depending on various factors such as length of the movie, director, actors, language, country of origin, year of release etc which is done by our machine learning system.

Literature Review

1) A hybrid approach for movie recommendation[1]

There are mainly two kinds of recommendation systems used for content recommendation (such as movies, music and many other scenarios), namely, content-based filtering and collaborative filtering. Both the methods have their strengths and weaknesses. Content-based recommendation systems use the metadata about movies such as title, user ratings, genre, cast, director, date of release and many other metrics to

present movies similar to what the user has selected or rated. On the other hand, collaborative filtering uses data about the ratings given by different users to the different movies on the platform. This method asks the user to rate a few movies; it then searches among the platform's users to find users with similar tastes and then recommends the movies liked by those users to the active user. Both types of techniques work well in different contexts, and both suffer in different contexts. This paper aims to make a hybrid recommendation system using both recommendation systems and switching between the two whenever deemed appropriate. This switching happens on criteria defined by the authors where the authors use collaborative filtering by default but switch to content filtering when the performance of collaborative filtering is poor. We also plan on implementing a hybrid recommendation system of sorts utilising the metadata about the movies as well as the user interaction data.

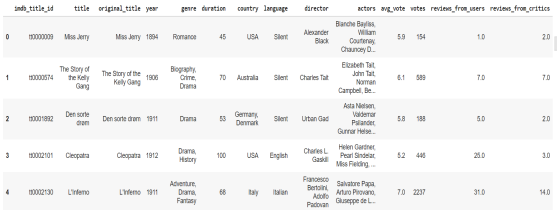
2) Movie Recommendation System Based on Movie Swarm[2]

In this paper, a new concept/algorithm was explored for movie recommendation systems - movie swarm mining. It uses two pruning rules and vertical data format frequent item mining. It solves the new item recommendation problem and provides an idea about the current trends of the popular movies and user interests. This is very helpful for movie producers to plan new movies. The paper claims that the proposed method is more efficient than the collaborative filtering method and content based recommendation system. The movie recommendation system mines movie databases to collect all the important information, such as, popularity and attractiveness, required for recommendation. The paper went on talking about how content based methods of recommendation or collaborative filtering fail at some point or the other - critical point of failure being unable to recommend to new users. However a hybrid of both these approaches,

though minises the issues, yet it faces difficulties in dealing with new users.

Dataset Description

(taken from: https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset)



	imdb_title_id	title	original_title	year	genre	duration	country	language	director	actors	avg_vote	votes	reviews_from_users	reviews_from_critics
0	tt0000009	Miss Jerry	Miss Jerry	1894	Romance	45	USA	Silent	Alexander Black	Blanche Bayliss, William Courtenay, Chauncey D...	5.9	154	1.0	2.0
1	tt0000574	The Story of the Kelly Gang	The Story of the Kelly Gang	1906	Biography, Crime, Drama	70	Australia	Silent	Charles Tait	Elizabeth Tait, John Tait, Norman Campbell, Be...	6.1	589	7.0	7.0
2	tt0001892	Den sorte drøm	Den sorte drøm	1911	Drama	53	Germany, Denmark	Silent	Urban Gad	Asta Nielsen, Valdemar Psilander, Gunnar Helse...	5.8	188	5.0	2.0
3	tt0002101	Cleopatra	Cleopatra	1912	Drama, History	100	USA	English	Charles L. Gaskill	Helen Gardner, Pearl Sindelar, Miss Fielding, ...	5.2	446	25.0	3.0
4	tt0002130	L'Inferno	L'Inferno	1911	Adventure, Drama, Fantasy	68	Italy	Italian	Francesco Bertolini, Adolfo Padovan	Salvatore Papa, Arturo Pirovano, Giuseppe de L...	7.0	2237	31.0	14.0

Our dataset contains the following description of the movie

- imdb_title_id: a unique id to uniquely identify the movie
- title: the current title of the movie (after remake/redub etc.)
- original_title: the title of the movie when it was released in its original format
- genre: the genre of the movie which can be one or more for each movie.
- country: the country in which movie was released
- language: the language in which movie was dubbed.
- director: Names of the director/s.
- actor: Names of the actors/actresses.
- avg_votes: average votes given to the movie
- reviews_from_users: number of users given by the users
- reviews_from_critics: number of users given by the critics

PreProcessing

Analysis on NULL values:

```
len(df_movies)
```

```
85855
```

```
df_movies.isnull().sum()
```

```
imdb_title_id              0
title                      0
original_title             0
year                       0
date_published             0
genre                      0
duration                   0
country                   64
language                 833
director                  87
writer                  1572
production_company      4455
actors                    69
description             2115
avg_vote                   0
votes                      0
budget                 62145
usa_gross_income       70529
worlwide_gross_income  54839
metascore              72550
reviews_from_users      7597
reviews_from_critics   11797
dtype: int64
```

Columns removed:

1. Description: reason is that it had fully textual data and would lead to NLP context which is not expected of the project.
2. production_company, budget, usa_gross_income, worldwide_gross_income, Metascore: columns with a high magnitude of null values are dropped as they cover a higher percentage of samples.

The rows with missing values are then finally dropped as there is no way of handling the data.

Outlier Detection

As we can see in the languages column of the movies, we can either drop them but that would cause a loss of data. What we observed is, that the given movies are very old and hence we can safely assume their language to being ‘Silent’ as initially, the movies did not contain any language. The screenshot of the same is shown below:

	imdb_title_id	title	original_title	year	genre	duration	country	language	
0	tt0000009	Miss Jerry	Miss Jerry	1894	Romance	45	USA	None	'
1	tt0000574	The Story of the Kelly Gang	The Story of the Kelly Gang	1906	Biography, Crime, Drama	70	Australia	None	Cl
2	tt0001892	Den sorte drøm	Den sorte drøm	1911	Drama	53	Germany, Denmark	NaN	L

0	tt0000009	Miss Jerry	Miss Jerry	1894	Romance	45	USA	Silent	Alexander Black	Blanche Bayliss, William Courtenay, Chauncey D...
1	tt0000574	The Story of the Kelly Gang	The Story of the Kelly Gang	1906	Biography, Crime, Drama	70	Australia	Silent	Charles Tait	Elizabeth Tait, John Tait, Norman Campbell, Be...
2	tt0001892	Den sorte drøm	Den sorte drøm	1911	Drama	53	Germany, Denmark	Silent	Urban Gad	Asta Nielsen, Valdemar Psilander, Gunnar Helse...

Now we had to look out for various genres, languages, actors, directors and countries as this is textual data and what machine understands is only numbers.
Hence we found out the number of distinct values in each of the following:

```
Unique values in genre 28
Unique values in country 214
Unique values in language 268
Unique values in director 25403
Unique values in actors 173433
```

We have used a bag of words implementation for the same as we cannot discard this data of text type as doing so would cause harm to the performance of the system.

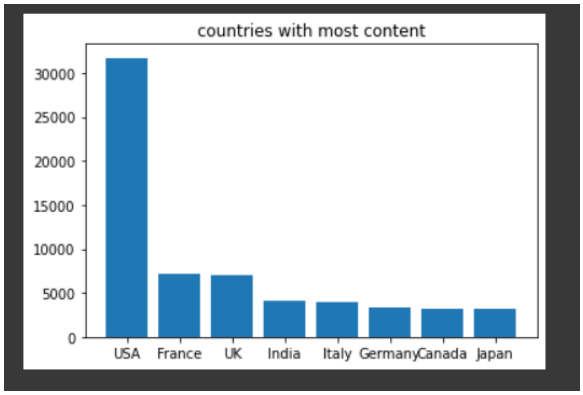
Different genres example:

```
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(df_movies['genre'])
vectorizer.get_feature_names()
```

```
['action',
 'adult',
 'adventure',
 'animation',
 'biography',
 'comedy',
 'crime',
 'documentary',
 'drama',
 'family',
 'fantasy',
 'fi',
 'film',
 'history',
 'horror',
 'music',
 'musical',
 'mystery',
 'news',
 'noir',
 'reality',
 'romance',
 'sci',
 'sport',
 'thriller',
 'tv',
 'war',
 'western']
```

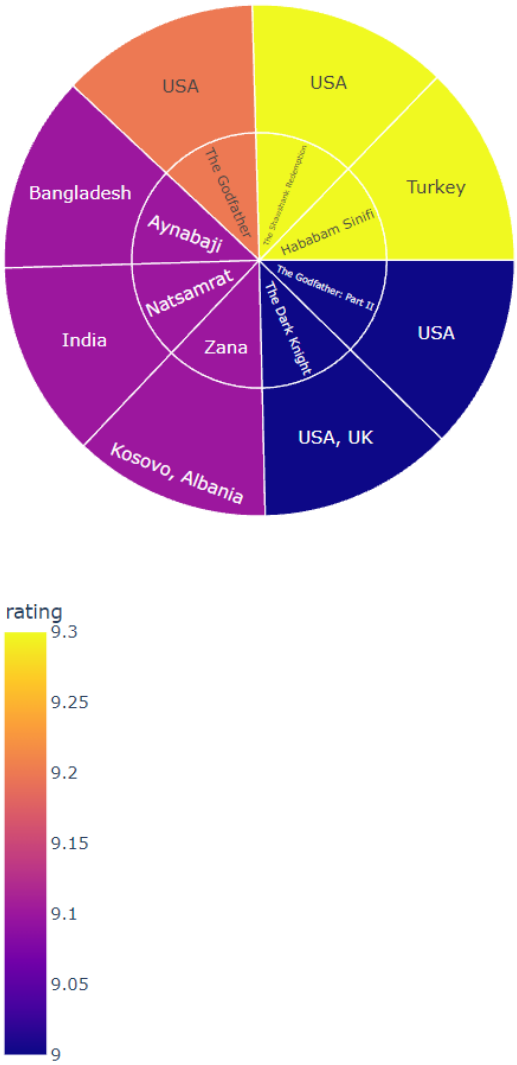
EDA

Countries with most content -

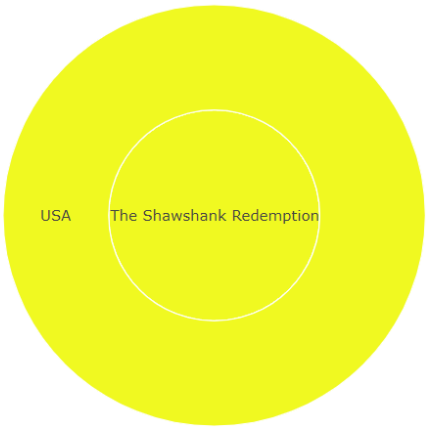


The content from the USA is extremely high than any other country as we can see in the above bar graph.

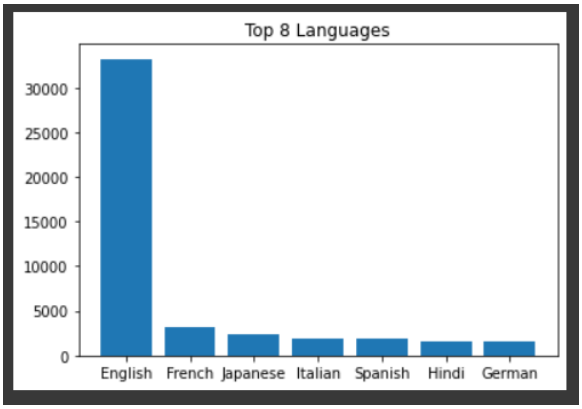
Most rated movies and their corresponding countries and average rating



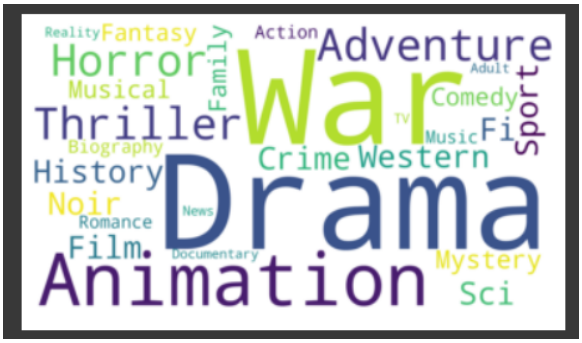
As we can see, the area in yellow has the highest avg rating and the movie is “Shawshank Redemption”. The graph also shows the country of the movies as well.



Top Movie Languages



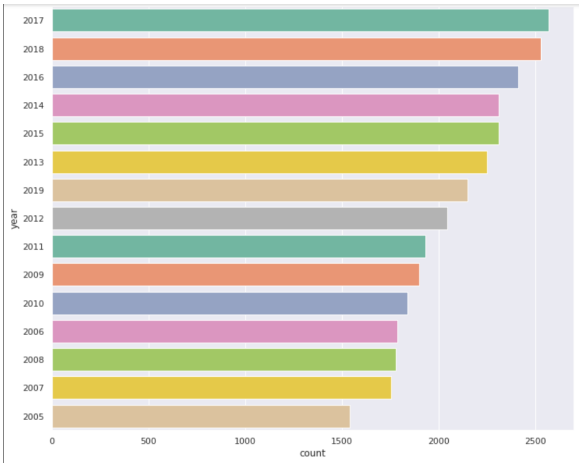
The number of movies in English is very high as compared to any other languages.



Word Cloud of All Genre

The size of each genre indicates the count of the movie with this particular genre. As we can see most of the movies were in the genre categories of Drama, Animation and War.

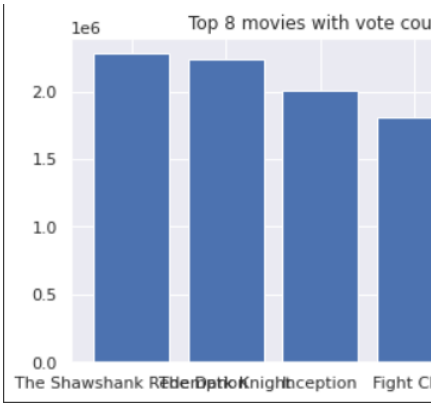
The number of movie release in each year



The above bar graph shows the number of movies that are released in a particular year. As

we can see the number of releases are more as we come near 2010 and is maximum in the 2017 year.

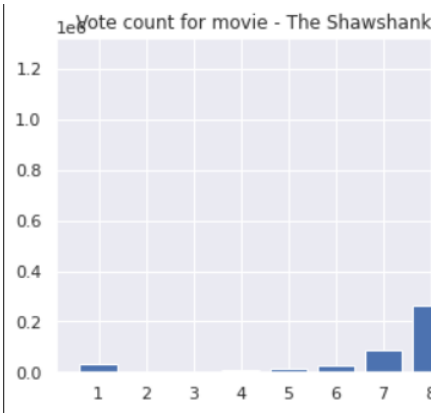
Most votes on a movie



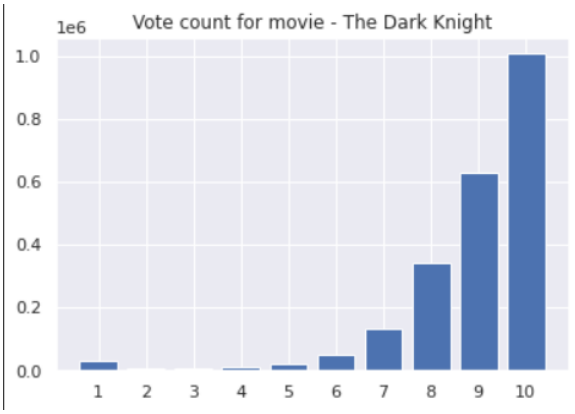
The above graph shows the top 5 movies on the basis of most number of votes.

Top Movies based on votes

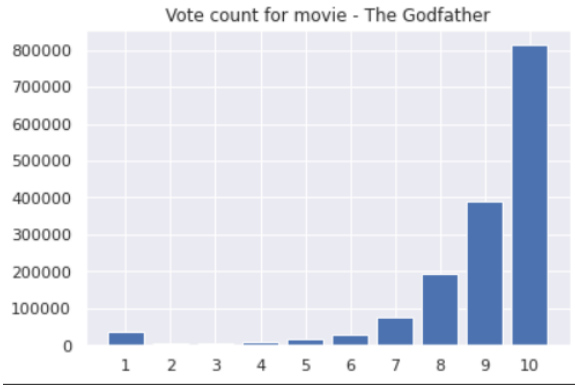
1.



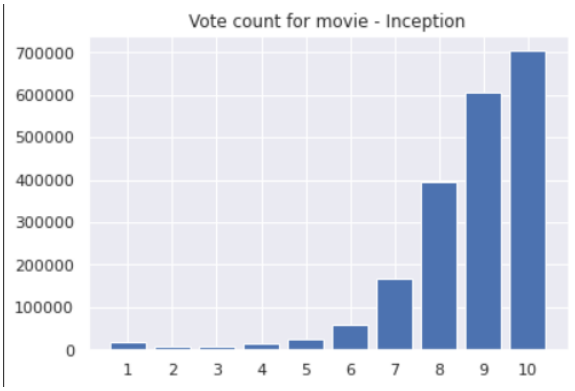
2.



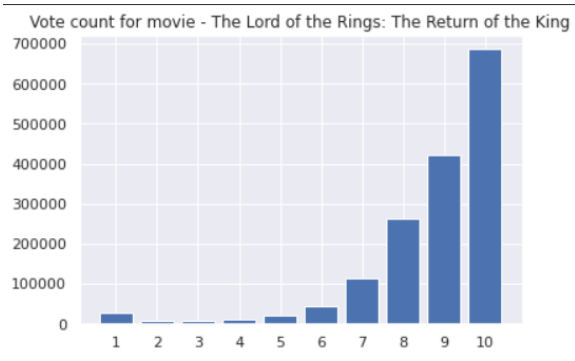
3.



4.



5.



The ranking is done by comparing first the number of votes_10 then votes_9 and so one. If any 2 movies has same number of votes_10 then the tie will be broken by the vote_9 votes and so on.

Methodology

We want to build a recommendation system that takes into account the preferences of a number of different users. We approach this problem from the viewpoint of content recommendations systems used by services like Amazon and Netflix.

There are three main approaches to content recommendation. The first is Demographic Filtering, where recommendations are based not on the user preferences but on the overall ranking of the movies based on global reviews received by the movies. For example, if we list the top 100 movies of all time on IMDB, the list will be the same for all users irrespective of their tastes or preferences. This technique is much better than randomly recommending movies and it is useful when you want to recommend content that is generally favoured by the larger population of viewers. But of course, this approach does not at all factor in user preference.

The second approach is Content Based Filtering. This technique uses the metadata about the movies such as movie title, synopsis, ratings, reviews, genre, release date, run time, cast, director and many other metrics, to find other movies which are similar to the movies selected by the user. When the user selects a movie, the movie is matched against other movies in the database to find the movies that are the most similar to the selected movie based on the above mentioned parameters. The top matches are then presented to the user and this is how content based filtering works in essence. The similarity metric being used can be something like cosine similarity, Pearson similarity or any other such metric. This approach does take into account the user's selection however it may fail to capture the general taste and inclination of the user. For instance, if the user selects a documentary, the user is most probably not just interested in watching documentaries and is also perhaps interested in science fiction action movies. This approach will most likely fail to capture this because documentaries may rank higher as per this approach.

The last approach is that of collaborative filtering. This approach does not require the movie metadata to make recommendations, instead it relies on the user ratings made by other users of the service. The idea with collaborative filtering is that we first collect ratings on some movies from the active user. We then find users in our database that are similar in their taste and preferences to the active user. We then use the movies preferred by these matched users to make recommendations to the active user. Hence, we are able to make predictions without using any metadata about the movies. This approach tries to match the general taste and behaviour of the active user to existing users in the database and then tries to predict what the active user might like. This method may also make novel recommendations to the user, which may differ quite a bit content-wise from the selected movies but still be rated high by the active user. This makes collaborative filtering a powerful technique. A major downfall of this approach, however, is that data sparsity can negatively impact its performance. The data about user ratings and reviews is likely to be sparse as not many users rate too many movies and the ratings of the active user may not overlap with any of the pre existing users in the database. Or the match might be made with a user who is in fact very dissimilar in terms of movie preferences from the active user. Thus, used on its own, the technique is prone to poor performance.

The idea with our approach is to combine some aspects of all three of the above techniques to produce an effective, robust and reliable recommendation system which uses both the movie metadata and the user reviews and rating data to make recommendations thus making full use of the available data. We want the three kinds of filtering to complement and support one another to make a hybrid recommendation system.

Model Details

For our baseline model, we have implemented a content filtering model that uses our preprocessed data to make movie recommendations. The model computes the Pearson Correlation between a movie of our choice and the rest of the movies in

the database. The model then ranks the movies in the database according to the Pearson Correlation values and presents the top 5 movies to the user. We have used Pearson Correlation instead of cosine similarity because Pearson correlation is indifferent to the absolute values of the metrics and it only considers the relative variations unlike cosine similarity. This is because of the way Pearson Correlation is formulated.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

We see that the mean is being subtracted from the data points. This eliminates the effects of the absolute mean values. Cosine similarity is not robust to these absolute value effects. If we take the mean to be zero, we see that the Pearson correlation degenerates into cosine similarity.

Results

```
Top 3 movies based on the ratings are -  Notuku Potu, Hopeful Notes, Jibon Theke Neya
```

Now we try to find the similar movies for these three movies.

1.
```
Notuku Potu
top 5 similar movies for  Notuku Potu are -
Notuku Potupearsonr value = 1.0
Alla älskar Alicepearsonr value = 0.9999954492433049
Il burberopearsonr value = 0.9999951928275973
God afton, Herr Wallenbergpearsonr value = 0.9999937410765766
Le battantpearsonr value = 0.9999931521555089
```

2.
```
Hopeful Notes
top 5 similar movies for  Hopeful Notes are -
Hopeful Notespearsonr value = 1.0
Machetepearsonr value = 0.9999990095796105
Y'en aura pas de facilepearsonr value = 0.9999985694655729
Riens du toutpearsonr value = 0.9999978875996978
La casa stregatapearsonr value = 0.9999970938295719
```

3.
```
Jibon Theke Neya
top 5 similar movies for  Jibon Theke Neya are -
Jibon Theke Neyapearsonr value = 0.9999999999999998
Utomlennye solntsem 2pearsonr value = 0.9999947153486807
Ramaleelapearsonr value = 0.9999915810761285
Govindudu Andari Vaadelepearsonr value = 0.9999877904309078
Punjab 1984pearsonr value = 0.9999866313746775
```

As we can see the best similar movie for let say movie X is X itself with the pearsonr value of 1 which means the exact similarity.

Also we can see the pearsonr value of each recommended movie.

Conclusion and learning

Learning from the project:

- team work

- how to handle large data
- EDA techniques
- how to make use of visualizations
- how to infer them to detect outliers
- how to handle null/nan values/missing values in the data
- how to represent textual data in the form of vectors
- which models to choose and how to choose an evaluation metric for unsupervised learning type of models.

Work left

- How to get better/efficient embeddings of the movies and what to neglect.
- How to choose a similarity metric for comparison between 2 movies.
- We have read about various techniques such as Pearson, cosine similarity, K-Means, KNN clustering, but we haven't yet dived into them
- as this part of model selection and tuning is scheduled for the later part of the semester.
- Given the results how to infer a relationship and what more to take into account while predicting.

Team member contribution

- Aniansh: Literature review and similarity metrics review.
- Bhavya: Data preprocessing and outlier detection.
- Harshal: Data collection and Literature review.
- Satwik: Exploratory data analysis.

References

[1] G. Lekakos and P. Caravelas, 'A hybrid approach for movie recommendation', *Multimed. Tools Appl.*, vol. 36, no. 1, pp. 55–70, Jan. 2008, doi: 10.1007/s11042-006-0082-7.

[2] 'Movie Recommendation System Based on Movie Swarm | IEEE Conference Publication | IEEE Xplore'. https://ieeexplore.ieee.org/document/6382910/authors#authors (accessed Oct. 12, 2021).