



# Factcheck: Real Time Detection of Misinformation

M Hema Sree, Sunija A P

School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India

[hemasree.m2020@vitstudent.ac.in](mailto:hemasree.m2020@vitstudent.ac.in)

**Abstract:** One of the biggest problems today is how fast misinformation spreads, affecting people, groups, and entire societies. With so many of all of the people relying on social media such as WhatsApp, Facebook, and blogs, it's more important now than ever it was to check for whether the information that we see is actually true. Misinformation can generate meaningful confusion as well as genuine harm, notably in countries like India, where getting reliable news stays important. That's precisely where this specific study comes directly in; the cool thing concerning our method is that it duly uses Machine Learning, Deep Learning, along with Natural Language Processing to quickly spot false information. This very system isn't just actually for certain researchers, it's specifically designed in order to help such media outlets, many policymakers, and even everyday people fight back against all of the fake news. And the key thing is that the model does quite well, showing an awesome 93.1% accuracy. Therefore, it becomes a great tool for the finding of misinformation.

**Keywords:** Machine Learning, Deep Learning, Natural Language Processing, Passive Aggressive Classifier, CatBoost, Random Forest, XGBoost, TextCNN, Misinformation, News, Prediction, Tf-idf Vectorization

## I. INTRODUCTION

Anyone may post anything on the internet these days, which has both advantages and disadvantages. The issue is that misleading information, particularly on social media, spreads like wildfire. Many individuals spread stories without even verifying their veracity, and before you realize it, they are causing miscommunications and even confrontations. The part that's crazy? In fact, false information has the power to sway public opinion and exacerbate hostility. However, technology has made it possible for us to create clever strategies for identifying and thwarting false news before it causes too much harm.

Mass media shapes how we see the world. No doubt about it. But here's the problem it's not always right. Sometimes, it gets twisted, used to spread lies, and that? That changes how people think. Fake news makes it tough. Hard to tell what's real, what's not. But here's the wild part! It's actually pretty good at catching lies. Research says AI can sift through fake news, slow it down before it spreads too far. A digital shield against deception.

The Fake news spreads fast. Too fast. Especially on social media, messaging apps places where a lie can go viral in minutes. And the damage? Real. Mob violence. Panic. People getting hurt, all because of something that wasn't even true. That's why this system exists.

It doesn't just find fake news. It calls it out. Labels it. Makes sure fewer people fall for it. The goal? Slow down the spread. Cut the impact. At its core, it's a text classifier sorting news into real or fake. Simple, but powerful. By verifying information, it stops lies before they take over. Helps people see the truth. Because in the end? That's what really matters.

## **II. RELATED WORK**

Misinformation detection is an unusual but interesting aspect of research because deep fakes are one of the largest concerns of our times. A click of a button can broadcast misinformation to millions of users and it becomes extremely complex to differentiate between reality and fiction. This is why innovative ways for detecting fake information on the internet have become the primary focus for many researchers. Numerous attempts have been made, from automated AI systems and fact-checking innovations, to tools designed for the dissemination of incorrect information:

Deep learning models mostly combining Long Short Term Memory networks for processing sequential data and feature extraction suggested in a 2019 paper on the identification of false news. The method concentrated on text data processing to efficiently detect false information. The research also recommended investigating hybrid models that blend different network designs for better performance and emphasized the possibility of integrating multi-modal data, such as text, photos, and videos.

Research conducted in 2021 used Graph Convolutional Networks (GCNs) to analyze user interactions and associations with news items to improve the identification of false news. Modeling user behavior and engagement patterns greatly increased the accuracy of detection. For a more thorough examination, future research attempts to broaden the model by including the temporal dynamics of user behavior and interactions.

The 2020 neural false news identification project aimed at spotting AI-generated bogus news made extensive pre-trained language models like GPT-2 and BERT usage. The study concentrated on identifying artificial intelligence (AI)-generated false material. To successfully control new and emerging kinds of disinformation, it highlighted the necessity of creating more resilient models that can adjust to quickly changing AI-generated material through adaptive learning.

To capture the temporal dependencies involved in the dissemination of false news, a model for identifying it on Twitter was created in 2022 utilizing Gated Recurrent Units, a form of recurrent neural network. To increase the accuracy of detecting false

information, the model attempted to improve detection by including a wider range of social cues, such as user behavior. Kapusta and Obonya (2020) focused on enhancing false news identification in using group analysis for morphologically reflective languages. Their study addressed challenges posed by complex word inflections, which often affect text-based models. By leveraging morphological structures, they improved the accuracy of misleading and fake news detection. The research demonstrated that language-specific preprocessing enhances classification performance. Their findings are valuable for fake news detection in linguistically rich languages.

### III. PROPOSED METHODOLOGY

This system's most notable feature is how straightforward it is. No complications. Everything is simply a process that functions. It begins with data collection: picking up the junk, and organizing the files. Then comes the most critical aspect extracting the key features that are significant. Training the Model. Teaching it to look for patterns and distinguish between truthful information and deception. And lastly? Putting it out there in the real world for action. Every step? Yes. There ain't a single needless action. Skip one, and everything collapses. This is how it maintains accuracy. Dependability. A system whose sole purpose is to prevent the widespread of false news.

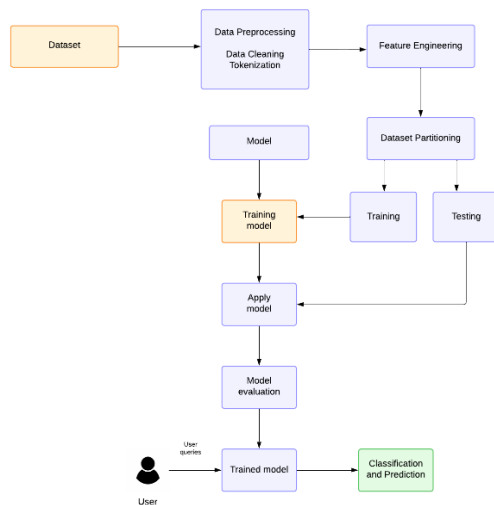


Fig. 1. Architecture

#### A. Data Collection

The fake news? It's cunning. It usually begins on dubious websites designed just to disseminate false information. And after an article has been published? It goes quickly. Sometimes individuals intentionally share it on social media, and other times they just

don't care to check. In any case, the harm has been done. Misinformation spreads like wildfire in this way. We used a dataset in order to investigate this issue. a blend of authentic and fraudulent news reports. 6335 data points—that's a lot of tales. Title, Text, Label, and one unidentified column are its four primary columns. Text and Title? The real news is found there. The Label? We can identify if it's real or fake based on that. Easy but effective.

## **B. Data Preprocessing**

The data was categorized into two sets: training and testing, and 20% of the data was reserved for testing. This is useful to verify how well the models perform. With sufficient samples, the division ensures that the models get to be trained on a sufficient portion of the data.

The very first order is removing unnecessary punctuation, special characters, and adding extra spaces, so that the process can be made smoother and more effective. Here we are removing the extra punctuation, special characters, and unnecessary spaces, which are the things that usually cause a disarray of the uniformity of classification making it hard to distinguish what is right from what is wrong. Lets hear it for tokenization, for the simple reason that tokenization provides the foundation for reducing large amounts of text to more manageable chunks. Furthermore, too common stop words are removed. Stopwords like the, is, and and are commonly used in text, but they play no evident function in distinguishing between authentic and bogus news. Additionally, lemmatization and stemming are two approaches that may be used to normalize words i.e., lemmatization can be considered as the method of evolving words in the same grammatical form whereas the stemmization removes the suffix from the word. For example, if we take the word "running" and apply lemmatization it will become "run", therefore the format becomes one for all different word forms of the same word. The data after that will be like new and will have a good structure and will create good features and training data sets. As a result, the preprocessing processes ensure that the model is both more accurate and less complicated, making the classification process far more efficient.

## **C. Feature Extraction**

A step that is extremely essential after preparation is feature extraction. This is when we transform words into numbers, so that models using machine learning can truly grasp and interpret them. One of the most effective ways to do this is TF-IDF vectorization and here's why it's so useful. TF-IDF (Term Frequency Inverse Document Frequency) is a unique approach to document analysis that goes beyond counting the number of times a word appears. Instead, it also looks at how common

that word is across the entire dataset. Another important thing is that TF-IDF is a way to filter out small words like "the," "is," or "and" that don't make much sense but to give emphasis to the words that do. Thus, machine learning models have an opportunity to be focused more on key terms that provide the best context and meaning.

$$TF-IDF_t = TF_t \times IDF_t = \frac{n}{k} \times \log \frac{D}{D_t}$$

Where:

$TF_t = n / k$ : Number of times the frequency of a term (t) in a text is calculated by dividing the total number of words (n) by the number of words (k).

$IDF_t = \log (D / D_t)$ : The total number of documents D divided by the number of documents  $D_t$  that use this phrase.

Now, when it comes to deep learning models like TextCNN, a different approach is preferred. Instead of doing the word frequency count method or simply counting words, word embeddings optimize the relationships between words into a multi-dimensional space by means of mapping them. This strategy is especially noteworthy in that it allows the model to comprehend context more efficiently. Hence the words that are close in denoting the meaning end up in the same spot on this space, consequently enabling the machine to detect patterns in the text more easily. For our model, we need to set up a few key parameters: `max_vocab_size`, `max_sequence_length`, `embedding_dim`. Just by the way, unlike in old-style methods, where each word is regarded as a separate unit, the word embeddings arrangement ranges the most similar words together. Instead of just storing words in a brain, the model has the knowledge of their interactions. This is quite impressive. Such a thing makes it so much easier to detect fake news. But before we feed words into the CNN, we should convert words into numbers. This is called tokenization. At first, we generate a Tokenizer and make it learn from the data. In the end, this generates a vocabulary and assigns each word with an own index. And then, we need to convert the text into sequences. Through this method, each word will be substituted with a simple number. Easy but enough problem. What about news items? Of course, not all of them are equal in length. Some of them are way too long, others are extremely short. In this way, we solve the matter by padding. If the article is insufficient, the text will be prolonged with several zeros. A balanced output. Consequently, the wordings become an organized input that the model uses to get the knowledge.

#### D. Model Training

When training models for misinformation detection, I experimented with both traditional machine learning models and a deep learning-based CNN model. The purpose was to realize how well the two methods can work in news sorting by separating true news from false news. As soon as we've cleaned up the data and highlighted the features with the help of TF-IDF vectorization, then comes the next task

that is to train the model. Machine learning is more of a concept here; it finds patterns in data to determine whether a news piece is true or not. To ensure that the model is capable of doing so, we divided the dataset into two parts: training (80%) and testing (20%). Now, here's what's interesting during training, the model studies real and fake news examples, learning to tell the difference. Through testing, we check how well it is doing on the data it has not encountered before. Another point that is noteworthy is the fact that we are not limited to the use of a single model. Rather, we apply a number of machine learning and deep learning algorithms to screen to select the most precise one. In this paper, we worked with a handful of the most popular models. Passive Aggressive Classifier, XGBoost, CatBoost and Random Forest. Next, I tried an ordinary CNN model, which is the best at catching patterns in the text. But the most interesting thing is that I introduced an improved version known as TextCNN. In any case, in order to compare their accuracy level it is important to test each of these models so that we can know which one will perform best.

#### **a. Passive Aggressive Classifier**

What's also interesting is that it stays passive when it correctly classifies data but aggressively adjusts itself when it makes a mistake. Because of this, it's particularly useful for binary classification problems especially when dealing with unbalanced datasets, where one group is much more common than the other. Additionally, it's worth considering that since misinformation evolves rapidly, having a model that adapts in real-time makes a huge difference. This is why the Passive Aggressive Classifier stands out in handling such challenges.

#### **b. XGBoost**

Gradient boosting is the technology that XGBoost is based on. This allows for better forecasts than a single strong model because it takes a few poor learners and merges them to build a very precise system. Another aspect that caught my eye is that XGBoost is incredibly fast and efficient and that is why it is frequently used on dealing with large datasets. Additionally, it is able to be optimized for missing values and be set up to run on lower memory, which is very helpful when working on very large text databases.

#### **c. CatBoost**

CatBoost, a boosting algorithm that has great potential for boosting algorithms in categorical data processing, stands out from many on the market. One of the most informative things to know is that it demands essentially little data preparation, which makes it an easier method different from those based on boosting to use. You should, however, also note that unlike many machine learning algorithms, CatBoost is precisely

designed to avoid overfitting, thus, it can maintain high accuracy rates even on complex datasets. To top it off, the ordered boosting method is used, which helps the algorithm perform at a high level and, thus, it is an ideal solution for NLP applications. So, CatBoost is a perfect and relevant solution when you are involved with textual data containing categorized features.

#### **d. Random Forest**

During training, the Random Forest ensemble learning technique creates many decision trees. After then, it combines all of their predictions to create a categorization that is more reliable and accurate. By averaging several predictions, this method reduces overfitting, an issue with individual decision trees. This enhances the ability to generalize to previously unknown news stories.

#### **e. TextCNN**

In this sense, the model captures the dynamics of text. The first stage extracts an embedding of words into dense vectors; the raw attention is on the convolutional layers that pick the essential textual features. Next, the layer of max-pooling makes sure we kill everything else and just focus on the important stuff in the end. Lastly, the model makes a binary classification. I used same padding instead of the standard padding, meaning the model preserves the original dimensionality of the input throughout. This will aid in retaining significant information from the text. It should also be noted that a dropout layer serves to counteract the overfitting. I trained the model using binary cross-entropy loss, as this is a binary classification challenge: real versus fake news. The optimizer? Adam would be a perfect choice for it because it's adaptive and works well across many problems. To determine if a model can distinguish between real and fake news, train it on the dataset and then evaluate its performance using important performance metrics including accuracy, precision, recall, and F1-score. The accuracy score, classification report, and model performance summary are generated.

### **E. Model Deployment**

Once all the models have been fully trained and tested, the best one is then selected for identification of misinformation in real-time. What's interesting is that this model is run in Flask, which a light web framework makes easy to plug machine learning into a web app. Here is a very interesting proposition that has user friendly interfaces. It has a simple input whereby a user types a news article in a few seconds gets an output: whether the news is real or fake. Thus, the model will assist the individual in quickly verifying information before sharing and will therefore become an invaluable resource in combating misinformation.

#### IV. RESULTS AND DISCUSSION

Passive Aggressive Classifier is the clear winner when put through an experimental analysis, scoring an accuracy of 93.1%. It is followed closely by XGBoost with a 92.2% accuracy score and CatBoost and Random Forest achieving 91% and 90.8%, respectively; textcnn follows closely at 91.7%. The results of the experiments show that Passive Aggressive Classifier outdid all others in terms of accuracy and efficiency. Although XGBoost and CatBoost performed excellently, they don't much favor the possibility of adapting in real time because of their processing complexity. And Random Forest, which is useful, dramatically lags behind accuracy. The approach TextCNN performed better in understanding context and nuanced textual patterns. They also found that the Passive-Aggressive Classifier not only achieved the highest accuracy but also adapts quite well to new input so that it can also be the best choice for misinformation detection. Hence, this model is best suited for real-time applications where news content needs instant verification. Its dynamic handling of misinformation gives it an edge over others, making detection more reliable in fast-changing environments.

The classifiers were trained and evaluated:

Model	Accuracy (%)
Passive Aggressive Classifier	93.1
XGBoost	92.2
TextCNN	91.7
CatBoost	91.0
Random Forest	90.8

Fig. 2. Accuracy

Furthermore, a classification report is produced that includes comprehensive metrics for every class, including accuracy, recall, and F1-score.

Accuracy: The overall correctness of the model's predictions.

$$A = \frac{TruePositive + TrueNegative}{TotalNumberofPredictions}$$

Precision: The proportion of actual positive predictions to all expected positives. It shows the proportion of chosen things that are pertinent.

$$P = \frac{TruePositive}{Positive + FalsePositive}$$

Recall (Sensitivity): The proportion of real positives to true positive predictions. It calculates the number of pertinent things chosen.

$$R = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}}$$

F1 Score: The harmonic mean of precision and recall, providing a balance between the two metrics.

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Among these, the Passive Aggressive Classifier demonstrated superior accuracy, leading to its selection for deployment.

Passive Aggressive Classifier classification report:

```

Accuracy: 0.931

Classification Report:

```

	precision	recall	f1-score	support
FAKE	0.93	0.93	0.93	628
REAL	0.93	0.93	0.93	639
accuracy			0.93	1267
macro avg	0.93	0.93	0.93	1267
weighted avg	0.93	0.93	0.93	1267

Fig. 3. Classification Report

Performance of the misinformation detection model was very good, scoring a noticeable accuracy of 93.1% in classifying real and fake news. Something interesting about the model, however, is that there is no serious imbalance in terms of false positives and true negatives. And even when looking at accuracies, recalls, or F1-scores, the results are undifferentiated at 0.93 for both REAL and FAKE news categories. Added with this is the reliability of the model calculated using weighted and macro averages both obtaining 0.93, showing that the model is overall consistent across different categories clearly highlighting effectiveness of the model in spotting misinformation. So, overall, the results surely say that this model will work best as an intelligent tool for disinformation detection since it can very effectively detect false and true news.

Confusion Matrix: A confusion matrix analysis further highlights the classification performance, showing high precision and recall for news detection.

It's also intriguing to observe how the confusion matrix basically offers more in-depth

understanding of the model's classification performance. It clearly shows the extent to which the model can differentiate real news from fake news. It has been able to correctly identify 585 cases of false news out of a total of 628, although 43 were misclassified as real. Similarly, it classified 595 cases of 639 correctly, but 44 cases were misclassified as fake. Another point of interest in these figures is that they seem to indicate a balanced error distribution, thus indicating that there is no favoritism towards one category or the other. As a result, it can be said that the model performs fairly and effectively in identifying false information. Also, the consistency in such a classification system reaffirms the model's reliability for the real-world application, which is highly essential when it comes to misinformation detection.

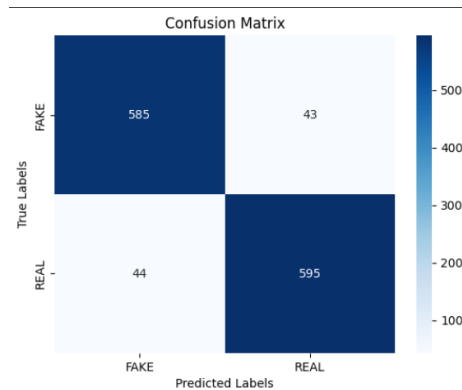


Fig. 4. Confusion Matrix

We can see how well the model distinguishes between fake and real news by looking at the confusion matrix. Although there are a few incorrect classifications, the accuracy is still quite high overall. The model's dependability for misinformation detection is validated by the confusion matrix.

## V. CONCLUSION

This research effectively creates a fast and precise misinformation detection system using machine learning, deep learning, and NLP. The approach is also intriguing. Initially, the team explored how misinformation spreads, its ramifications, and various ways to detect it. Next, a procedure was developed that processes news articles by cleaning the text, applying stemming and TF-IDF for retrieving useful features for detecting fraud news. The use of a textcnn model, which employs an embedding layer to transform words into dense vectors before undergoing convolutional processes for crucial textual feature identification, came next and through a max-pooling layer focusing on the most salient features, leading to a final output in binary classification. Subsequently, the training of a model for classifying news articles that boast awesome

levels of accuracy was accomplished-an impressive 93% in differentiating between real and false news. The best model, Passive Aggressive Classifier, was later embedded in a real-time Flask application, thus offering users the opportunity to access an instant verification of news articles making it more of a useful tool for daytoday people rather than an experimental aspect of research. Another area to mention for the future thought includes possible improvements. Accelerating the creation of a mobile application for quick fact-checking is a crucial area for the future. It is also worth considering expanding the dataset for enhancing robustness and deployment of modern deep learning models to impart better contextual understanding. This additionally leads to another exciting prospect -the implementation of explainability techniques could go a long way towards showing the user why an article is marked as real or fake, thereby instilling confidence in the system. In conclusion, the findings suggest that machine learning algorithms offer a powerful, scalable solution to the problem of misinformation. By continuing to strengthen with ongoing improvements, along with advanced AI integration techniques, a more transparent and dependable real-time fact-checking system can be developed.

## VI. REFERENCES

- [1] Beatriz Villarejo-Carballido, Gisela Redondo-Sama, Laura Ruiz-Eugenio, and Cristina M. Pulido. A novel use of social effect on social media to combat health-related bogus news. *Public health and environmental research internationaljournal*, 2020.
- [2] Shu and colleagues (2017). A Data Mining Approach to Social Media Fake News Detection.
- [3] Bhowmik, D., Zargari, S., and Ajao, O. (2018). Twitter Fake News Detection Using Hybrid CNN and RNN Models
- [4] DSKR Rupanjal Dasgupta and Vivek Singh. automated identification of false information by machine learning and language analysis.
- [5] F. Monti and associates (2019). Geometric Deep Learning for Social Media Fake News Detection.
- [6] T. Castelo and colleagues (2019). Topic-Agnostic Fake News Identification: A Thorough Examination of Social Features and Content.
- [7] An Overview of Fake News: Essential Theories, Identifying Techniques, and Prospects, Zafarani, R., and Zhou, X. (2020).
- [8] Liu, H., Wang, S., Tang, J., Shu, K., and Sliva, A. (2017). A Data Mining Approach to Social Media Fake News Detection.
- [9] Saba-Sadiya, S., Karimi, H., Roy, P., & Tang, J. (2018). Multi-source, multi-class detection of fake news.
- [10] Wu, Y. F. B., and Liu, Y. (2018). Early Social Media Fake News Identification Using Recurrent Networks and Propagation Path Classification
- [11]In 2021, Alenezi, M., and Alqenaeci, S. Is it a fake? Automated identification of

- false and misleading material about COVID-19 in digital media and social networks
- [12] Media-rich false news detection: A survey by S. B. Parikh and P. K. Atrey, Proceedings of the IEEE Conference on Multimedia Information Process Retr. (MIPR), April 2018, pp. 436–441.
- [13] False information detection in online material and its function in decision making: A comprehensive literature review, Social Network Analytic Mining, vol. 9, no. 1, pp. 1–20, December 2019. A. Habib, M. Z. Asghar, A. Khan, and A. Khan.
- [14] Meel and Vishwakarma, P., Fake news, rumors, and information pollution in the web and social media: A current overview of the latest developments, obstacles, and possibilities, Expert Syst. Appl., vol. 153, Sep. 2020, Art. no. 112986.
- [15] Using emotional cues to identify believability, by A. Giachanou, P. Rosso, and F. Crestani, in Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., July 2019, pp. 877–880.
- [16] Inf. Sci. Impact, Res. Community, vol. 52, no. 1, pp. 1–4, 2015; Conroy, V. L. Rubin, and Y. Chen, Automatic deception detection: Techniques for identifying bogus news.
- [17] Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI): Detecting rumors from microblogs with recurrent neural networks, J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha. Inf. Syst. Res. Collection School, 2016, pp. 3818–3824.
- [18] In Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (ACL), J. Ma, W. Gao, and K.-F. Wong use propagation structure via kernel learning to identify rumors in microblog entries. Res. Collection School Comput. Inf. Syst., July/Aug. 2017, pp. 708–717, Vancouver, BC, Canada.
- [19] "Bend the truth": Benchmark dataset for false news identification in Urdu and its assessment, J. Intell. Fuzzy Syst., vol. 39, no. 2, pp. 2457–2469, 2020; M. Amjad, G. Sidorov, A. Zhila, H. Gómez-Adorno, I. Voronkov, and A. Gelbukh.
- [20] Enhancing false news detection using domain-adversarial and graph-attention neural networks, H. Yuan, J. Zheng, Q. Ye, Y. Qian, and Y. Zhang, Decis. Support Syst., vol. 151, Dec. 2021, Art. no. 113633.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

