**Title: Thematic development of cybersecurity pre and post covid**

**SECTION 1 Introduction :**

The world has faced an unprecedented crisis of the covid-19 pandemic. It has altered the lives of billions of people and changed the new normal in terms of the way we live and work. Aside from the extraordinary impact on society as a whole, the pandemic has brought about a massive shift towards digitalization. In a short span of time, educational institutes, businesses- small and big, healthcare services, all went online, bringing about a revolution of sorts. This revolution in digitalization also meant the advent of new challenges, specifically in the areas of cybersecurity. While this did prevent the world from coming to a standstill, it made us more vulnerable to a wide range of cyber attacks like phishing scams, data and privacy breaches, malware attacks to name a few and has made cybersecurity more vital than it has ever been.

Objectives of this paper -

1. Identifying which topics were researched upon by the scientific as well as the non-scientific community in the domain of cybersecurity.
2. Analyze trends in topics researched pre vs post covid.
3. Year-wise trend analysis in topics researched in the cybersecurity domain.

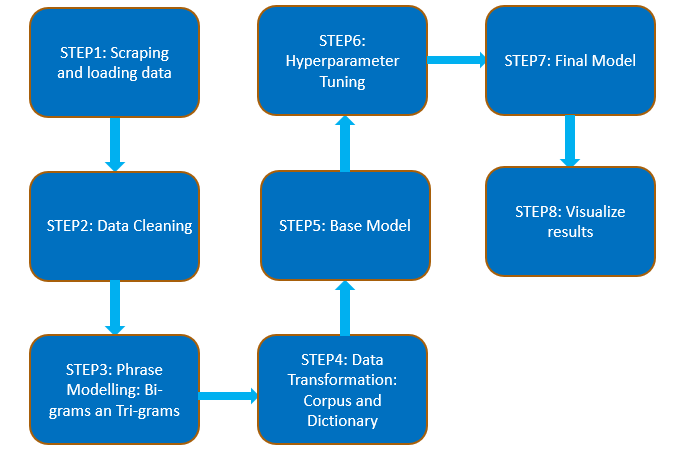
We perform topic modeling on our dataset using Latent Dirichlet Allocation (LDA)

**Topic modeling** is a type of statistical modeling for discovering the abstract “topics” that occur in a collection of documents. It is an unsupervised machine learning method that helps us discover hidden semantic structures in a paper, that allows us to learn topic representations of papers in a corpus.

**Latent Dirichlet Allocation** (LDA) is a method of topic modeling and is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions. We are going to apply LDA to a set of documents and split them into topics.

In this paper, we’ll go a few steps deeper by outlining the framework to quantitatively evaluate topic models through the measure of topic coherence.

**Block diagram**

****

We collect the abstract for peer-reviewed literature and article content for non-peer-reviewed literature. Next, we clean the text by removing unwanted characters, lemmatizing, tokenization, and removing stopwords/unwanted words. We then go on to identify phrases of sizes 2 and 3 (bi-grams and tri-grams) that the topic model can recognize. We filter bi-grams and tri-grams for noun structures and concatenate these phrases into a word. The next step involves creating a dictionary and corpus data structure from the bag of words and passing it to the LDA algorithm to obtain a base model. We evaluate the topic model using the c\_v coherence score. The base model is tuned with different hyperparameters by comparing the coherence scores and the optimal set of parameter values is used to obtain our final model. The final LDA model provides us with keywords for each topic and it also assigns weight to these keywords. Finally, we manually draw inferences for these topics and visualize these results.

**SECTION 3 Methodology: Literature review**

Peer review is an academic term for quality control. Each article published in a peer-reviewed journal is closely examined by a panel of reviewers who are experts on the article's topic. The reviewers assess the author’s proper use of research methods, the significance of the paper’s contribution to the existing literature, and check on the authors’ works on the topic in any discussions or mentions in citations.

<https://clarksoncollege.libanswers.com/faq/236975>

Peer-reviewed literature is also sometimes referred to as scholarly literature. The peer-review process subjects an author's scholarly work, research, or ideas to the scrutiny of others and is considered necessary to ensure academic scientific quality.

Non-Peer Reviewed literature isn't usually written by experts on their topics. While they do undergo review by an editor, they don’t receive peer review and are often biased to some degree. Book reviews, blogs, and news articles are examples of peer-reviewed publications.

<https://www.usgs.gov/faqs/what-does-it-mean-when-publication-peer-reviewed>

We explored 3 databases-

1. Arxiv - Collected a dataset of 600 research papers with the domain computer science and keyword "cybersecurity” and ran our preliminary analysis on it (We have the results after running LDA and analyzing the topics obtained, however, the process we used there wasn’t refined, didn’t use n-grams or hyperparameter tuning) Due to the nature of papers available on Arxiv (because such papers “preprints” and are not equivalent to publications in a peer-reviewed journal) and limitation of number of papers available we shifted to Scopus for peer-reviewed literature.

<https://arxiv.org/>

2. Scopus - Obtained our peer-reviewed corpus from Scopus by filtering for papers having the keyword “cybersecurity” in them. We collected a corpus of 10680 papers, 6680 pre covid, and 4000 post covid. Collected documents year-wise, for ease of plotting year-wise trends. The data exported contained Date, Title, and Abstract.

<https://www.scopus.com/>

A more elaborate count -

| **YEAR** | **COUNT** |
| --- | --- |
| 2010 | 107 |
| 2011 | 141 |
| 2012 | 168 |
| 2013 | 229 |
| 2014 | 305 |
| 2015 | 504 |
| 2016 | 734 |
| 2017 | 1030 |
| 2018 | 1461 |
| 2019 | 2000 |
| 2020 | 2000 |
| 2021 | 2000 |

Reasons for the following count-

There was a limitation of extracting a maximum of 2000 papers at once from Scopus. Hence, for the years that had more than 2000 papers published, we exported the top 2000 papers based on their citation count. For the years prior from 2000 to 2009 Scopus only had 241 papers, hence these papers were excluded from the analysis. We tried to extract a balanced corpus for pre and post covid papers and also collected more papers around the time of covid (for 2018-2019 and 2020-2021) to be able to show a trend in topics due to covid.

3. Collected non-peer-reviewed literature by scraping three popular cybersecurity blog websites - Krebson security, Schneier, and Lastwatchdog. A total of 11394 articles were scraped - 10150 articles pre covid and 1244 articles post covid.

<https://krebsonsecurity.com/>

<https://www.lastwatchdog.com/>

<https://www.schneier.com/>

Reason for inclusion - we included peer-reviewed and non-peer-reviewed literature as well to show the differences between the topics researched by the scientific community (experts in the domain) and the non-scientific community. We chose the most popular blog websites according to a report by chub in 2021

<https://www.cshub.com/executive-decisions/articles/the-top-ten-cyber-security-blogs>

Extracted all the articles from these three websites. Didn’t scrape more websites to keep a balanced corpus for peer-reviewed and non-peer-reviewed literature and also because of the limitation of the long running time of the algorithm. (for a large corpus, the time complexity of LDA is exponential)

A more elaborate count -

| **WEBSITE** | **PRE-COVID** | **POST-COVID** |
| --- | --- | --- |
| Krebson | 1730 | 270 |
| Watchdogs | 1062 | 354 |
| Schneier | 7358 | 620 |

**SECTION 4 Algorithm and Analysis:**

**A] Each block explained in detail - (of figure in introduction)**

**Step 1: Scraping and loading data**

A Peer-reviewed corpus of 10680 papers was collected from Scopus and the title and abstract of the papers were loaded in a pandas dataframe.

Non-peer-reviewed literature was collected by scraping three popular cybersecurity blog websites - Krebson security, Schneier, and Lastwatchdog. A total of 11394 articles were scraped. The title and article’s content were loaded in a pandas dataframe.

**Step 2: Data Cleaning**

Machine Learning needs data in numeric form. We basically used an encoding technique (BagOfWords in our case) to encode text into numeric vectors. But before encoding, we first need to clean the text data and this process to prepare (or clean) text data before encoding is called text preprocessing, this is the very first step to solve the NLP problems.

We perform a simple preprocessing on the content of abstracts of papers to get more reliable results. To do that, we’ll use a regular expression to remove any punctuation and URLs and then lowercase the text (removing URLs and unwanted characters are crucial as it would give us unwanted keywords in the topics extracted). We then tokenize each sentence into a list of words. Another step in data cleaning is removing stop words (A stop word is a commonly used word such as “the”, “a”, “an”, “in” etc) and lemmatization, which transforms words to their most basic form, such as ‘running’ and ‘ran’ to ‘run’ so that they are recognized as the same word. (these two steps have been performed along with phrase modeling though)

**Step 3: Phrase Modeling**

We want to identify phrases so the topic model can recognize them. Bigrams are phrases containing 2 words e.g. ‘social media’. Likewise, trigrams are phrases containing 3 words e.g. ‘Proctor and Gamble’.

There are many ways to detect n-grams, explained [here](http://bit.ly/2HGWhl8). We shall use the Pointwise *Mutual Information (PMI) score.* This measures how much more likely the words co-occur than if they were independent. The metric is sensitive to rare combinations of words, so it is used with an occurrence frequency filter to ensure phrase relevance.

We filter bigrams or trigrams with noun structures (higher n-grams are not used because phrases of sizes greater than 3 have a significantly lower chance of being repeated multiple times). This helps the LDA model better cluster topics, as nouns are better indicators of a topic being talked about. We use the NLTK package to tag part of speech and filter these structures. Nouns give more information about the topic whereas other parts of speech give more explanation about the topic. Hence we filter the text to provide us with more noun structures.

Before running our LDA model we remove words such as cybersecurity, article, paper, research, and study. These words are removed because they are redundant with respect to the keyword used to search the papers. If this step is not done, LDA may create a topic that has these keywords and may put a large number of papers under this topic.

https://towardsdatascience.com/6-tips-to-optimize-an-nlp-topic-model-for-interpretability-20742f3047e2

**Step 4: Data transformation: Corpus and Dictionary**

The two main inputs to the LDA topic model are the dictionary(id2word) and the corpus.

The dictionary is a mapping of a unique id for each word in the dataset. Gensim library creates a unique id for each word in the document. The obtained corpus is a mapping of word\_id and word\_frequency in each document.

**Step 5: Base Model**

Now in order to run our LDA model using Gensim, we need to provide the dictionary, corpus, and values of the hyperparameter. The LDA model provides us with keywords for each topic and it also assigns weight to these keywords. These weights signify the importance of the keywords.

The need for evaluating topic models -

We know probabilistic topic models, such as LDA, are popular tools for text analysis, providing both a predictive and latent topic representation of the corpus. However, there is a longstanding assumption that the latent space discovered by these models is generally meaningful and useful and that evaluating such assumptions is challenging due to its unsupervised training process. Besides, there is a no-gold standard list of topics to compare against every corpus.

Nevertheless, it is equally important to identify if a trained model is objectively good or bad, as well as to have an ability to compare different models/methods. To do so, one would require an objective measure for the quality. Traditionally, and still for many practical applications, to evaluate if “the correct thing” has been learned about the corpus, an implicit knowledge and “eyeballing” approaches are used. Ideally, we’d like to capture this information in a single metric that can be maximized, and compared.

Some approaches that are commonly used for evaluation:

Eyeballing Models

* Top N words
* Topics / Documents

Intrinsic Evaluation Metrics

* Capturing model semantics
* Topics interpretability
* Perplexity and coherence scores can be used to quantitatively evaluate the model on the two intrinsic evaluation metrics.

Human Judgements

* What is a topic

Extrinsic Evaluation Metrics/Evaluation at task

* Is the model good at performing predefined tasks, such as classification

<https://datascienceplus.com/evaluation-of-topic-modeling-topic-coherence/>

In this paper, we’ll use topic coherence, an intrinsic evaluation metric, to quantitatively justify the model selection.

<https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0>

Before we used topic coherence, we briefly explored perplexity measure. Perplexity as well is one of the intrinsic evaluation metrics and is widely used for language model evaluation. It captures how surprised a model is of new data it has not seen before and is measured as the normalized log-likelihood of a held-out test set. Focussing on the log-likelihood part, you can think of the perplexity metric as measuring how probable some new unseen data is given the model that was learned earlier. That is to say, how well does the model represent or reproduce the statistics of the held-out data. However, recent studies have shown that predictive likelihood (or equivalently, perplexity) and human judgment are often not correlated, and even sometimes slightly anti-correlated. Hence optimizing for perplexity may not yield human interpretable topics.

This limitation of perplexity measure thus requires us to use a different metric to model the human judgment, and thus Topic Coherence. The concept of topic coherence combines a number of measures into a framework to evaluate the coherence between topics inferred by a model.

TOPIC COHERENCE -

Topic Coherence measures score a single topic by measuring the degree of semantic similarity between high scoring words in the topic. These measurements help distinguish between topics that are semantically interpretable topics and topics that are artifacts of statistical inference.

A set of statements or facts is said to be coherent if they support each other. Thus, a coherent fact set can be interpreted in a context that covers all or most of the facts. An example of a coherent fact set is “the game is a team sport”, “the game is played with a ball”, “the game demands great physical efforts”

COHERENCE MEASURES

A brief look at different coherence measures, and how they are calculated:

* **c\_v** measure is based on a sliding window, one-set segmentation of the top words, and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity
* **c\_p** is based on a sliding window, one-preceding segmentation of the top words, and the confirmation measure of Fitelson’s coherence
* **c\_uci** measure is based on a sliding window and the pointwise mutual information (PMI) of all word pairs of the given top words
* **c\_umass** is based on document co-occurrence counts, a one-preceding segmentation, and a logarithmic conditional probability as confirmation measure
* **c\_npmi** is an enhanced version of the C\_uci coherence using the normalized pointwise mutual information (NPMI)
* **c\_a** is based on a context window, a pairwise comparison of the top words, and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity

**We choose c\_v measure to evaluate the topic models:**

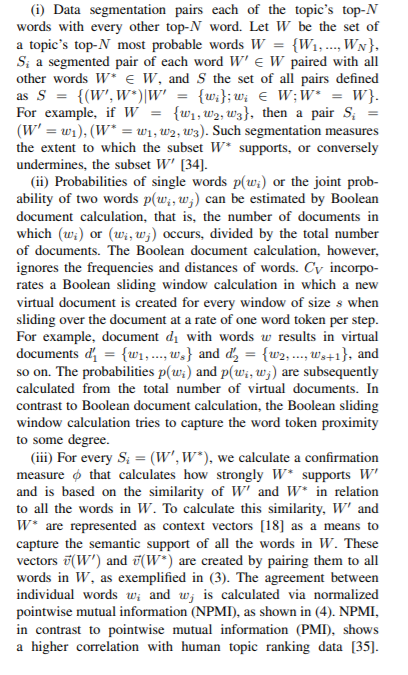
A simple explanation of c\_v -

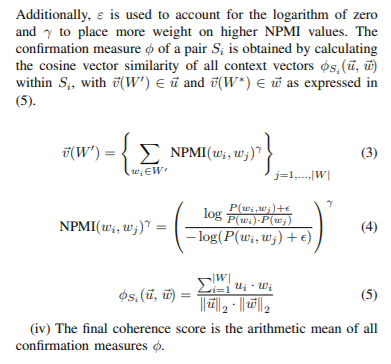
c\_v is based on a sliding window, a one-set segmentation of the top words, and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity. This coherence measure retrieves co-occurrence counts for the given words using a sliding window and the window size 110. The counts are used to calculate the NPMI of every top word to every other top word, thus, resulting in a set of vectors—one for every top word. The one-set segmentation of the top words leads to the calculation of the similarity between every top word vector and the sum of all top word vectors. As a similarity measure, the cosinus is used. The coherence is the arithmetic mean of these similarities.

A deeper mathematical explanation adapted from:

<https://ieeexplore.ieee.org/document/8259775>

This paper adopts the c\_v coherence measure for topic coherence calculations. c\_v is based on four parts: (i) segmentation of the data into word pairs, (ii) calculation of word or word pair probabilities, (iii) calculation of a confirmation measure that quantifies how strongly a word set supports another word set, and finally (iv) aggregation of individual confirmation measures into an overall coherence score.





(We haven’t made use of any of the formulas mentioned, we simply used a library - gensim that calculates this for us)

<https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0>

**Step 6: Hyperparameter Tuning**

The accuracy of the LDA model (coherence value) can be improved by tuning the hyperparameters: the number of topics, chunk size, passes, and iterations. A brief explanation of each of the hyperparameters is given below.

* **Chunksize**: the number of documents to be loaded into memory each time for training.
* **Passes**: the number of training iterations through the entire corpus.
* **Iterations**: the maximum iterations over each document to reach convergence — limiting this means that some documents may not converge in time.
* **Number of topics (T):** the count of topics to be extracted from the corpus.

The topic distribution for the entire corpus is updated after each chunksize, and after each pass. Increasing chunksize to the extent the memory can handle will increase speed as topic distribution update is expensive. However, increasing chunksize requires increasing the number of passes to ensure sufficient corpus topic distribution updates, especially in small corpora. Iterations also need to be high enough to ensure a good amount of documents reach convergence before moving on. A large corpus shall contain a number of topics, having a small T would club relatively unrelated topics into a single topic on the other hand having a large T may cluster the topics (divide one topic into a number of closely related topics) and give us less interpretable topics.

<https://papers.nips.cc/paper/2010/file/71f6278d140af599e06ad9bf1ba03cb0-Paper.pdf>

In our analysis we have optimized the hyperparameters linearly (i.e. one hyperparameter at a time), this was done to reduce the time taken to tune the hyperparameters. Optimizing the parameters with 4 for loops would have been ideal, but it would have taken weeks or even months to compute the optimal coherence score.

The hyperparameters were tuned in the following order: chunksize, passes, iterations, and the number of topics.

**Step 7: Final Model**

After hyperparameter tuning, we get the optimal set of hyperparameter values for our LDA model. These values are used to obtain the set of topics for our corpus. The topic names are manually inferred from the set of keywords for each topic.

**Step 8: Visualize results**

We use four ways to visualize the results of our final models. Firstly, PyLDAvis is used to visualize the topics of our final model. Second, we visualize the topics using a word cloud. Third, we represent the proportion of papers for each topic in the pre and post covid period using stacked bar graphs. Two such bar graphs are displayed in our analysis. Lastly, a year-wise analysis is also performed for each topic to show the percentage of papers belonging to a particular topic in a given year.

**The algorithm; its pitfalls; and choice of selection:**

# **Different Methods of Topic Modeling**

* Latent Dirichlet Allocation (LDA)
* Non Negative Matrix Factorization (NMF)
* Latent Semantic Analysis (LSA)
* Parallel Latent Dirichlet Allocation (PLDA)
* Pachinko Allocation Model (PAM)

<https://iq.opengenus.org/topic-modelling-techniques/>

Latent Dirichlet Allocation (LDA) is a generative statistical model that helps pick up similarities across a collection of different data parts. In topic modeling, each data part is a word document (e.g. a single abstract of a research paper) and the collection of documents is a corpus (e.g. all abstracts of the collected research papers). Similar sets of words occurring repeatedly may likely indicate topics.

LDA has made a big impact in the fields of natural language processing and statistical machine learning and has quickly become one of the most popular probabilistic text modeling techniques in machine learning.

An advantage of the LDA technique is that one does not have to know in advance what the topics will look like. By tuning the LDA parameters to fit different dataset shapes, one can explore the topic formation and resulting document clusters.

<https://www.airccj.org/CSCP/vol6/csit65316.pdf>

LDA assumes that each document is represented by a distribution of a fixed number of topics, and each topic is a distribution of words.

Algorithm’s simplistic key steps to approximate these distributions:

* User selects K, the number of topics present, tuned to fit each dataset.
* Go through each document, and randomly assign each word to one of K topics. From this, we have a starting point for calculating document distribution of topics p(topic t|document d), the proportion of words in document d that are assigned to topic t. We can also calculate the topic distribution of words p(word w|topic t), the proportion of word w in all documents’ words that are assigned to topic t. These will be poor approximations due to randomness.
* To improve approximations, we iterate through each document. For each document, go through each word and reassign a new topic, where we choose topic t with a probability p(topic t|document d) ∗p(word w|topic t) based on the last round’s distribution. This is essentially the probability that topic t generated word w. Recalculate p(topic t|document d) and p(word w|topic t) from these new assignments.
* Keep iterating until topic/word assignments reach a steady-state and no longer change much, (i.e. converge). Use final assignments to estimate topic mixtures of each document (% words assigned to each topic within that document) and the word associated to each topic (% times that word is assigned to each topic overall).

A deeper mathematical explanation of the algorithm-

Adapted from: <https://ai.stanford.edu/~ang/papers/jair03-lda.pdf> Page 3-4

**Notation and terminology**

Formally, we first define the following terms:

• A word is the basic unit of discrete data, defined to be an item from a vocabulary indexed by {1,...,V}. We represent words using unit-basis vectors that have a single component equal to one and all other components equal to zero. Thus, using superscripts to denote components, the Vth word in the vocabulary is represented by a V-vector w such that w^v = 1 and w^u = 0 for u != v.

• A document is a sequence of N words denoted by w = (w1,w2,...,wN), where wn is the nth word in the sequence.

• A corpus is a collection of M documents denoted by D = {w1,w2,...,wM}.

We wish to find a probabilistic model of a corpus that not only assigns high probability to members of the corpus but also assigns high probability to other “similar” documents.

**Latent Dirichlet allocation**

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.

LDA assumes the following generative process for each document w in a corpus D:

1. Choose N ∼ Poisson(ξ).

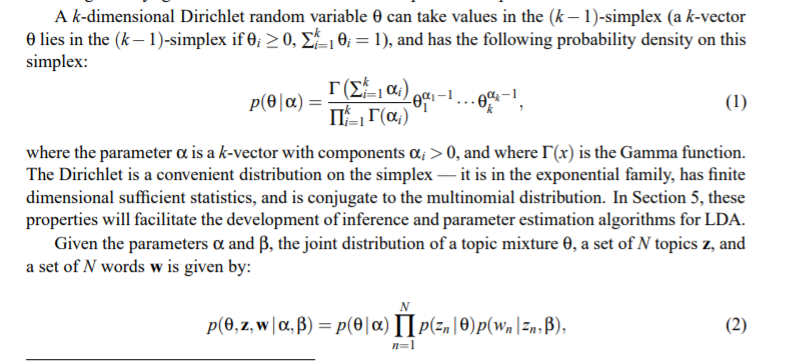
2. Choose θ ∼ Dir(α).

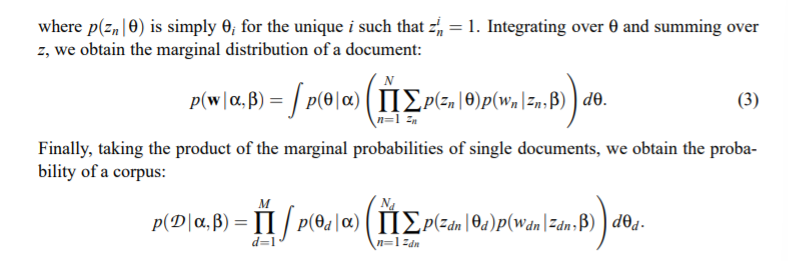
3. For each of the N words wn:

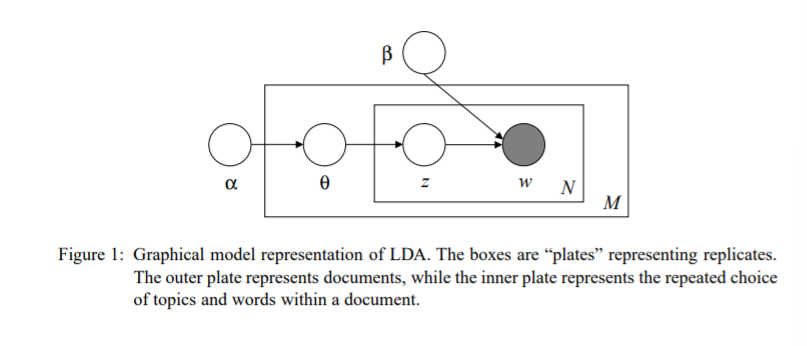
1. Choose a topic zn ∼ Multinomial(θ).
2. Choose a word wn from p(wn |zn,β), a multinomial probability conditioned on the topic zn.

**Assumptions**

Firstly, the dimensionality k of the Dirichlet distribution (and thus the dimensionality of the topic variable z) is assumed known and fixed. Second, the word probabilities are parameterized by a k ×V matrix β where βij = p(w^j = 1| z^i = 1), we treat it as a fixed quantity. Finally, the Poisson assumption is not critical to anything that follows and more realistic document length distributions can be used as needed. Furthermore, note that N is independent of all the other data-generating variables (θ and z).







The LDA model is represented as a probabilistic graphical model in Figure 1. There are three levels to the LDA representation. The parameters α and β are corpus level parameters, assumed to be sampled once in the process of generating a corpus. The variables θd are document-level variables, sampled once per document. Finally, the variables zdn and wdn are word-level variables and are sampled once for each word in each document.

It is important to distinguish LDA from a simple Dirichlet-multinomial clustering model. A classical clustering model would involve a two-level model in which a Dirichlet is sampled once for a corpus, a multinomial clustering variable is selected once for each document in the corpus, and a set of words are selected for the document conditional on the cluster variable. As with many clustering models, such a model restricts a document to being associated with a single topic. LDA, on the other hand, involves three levels, and notably, the topic node is sampled repeatedly within the document. Under this model, documents can be associated with multiple topics.

**Complexity analysis**

For a small number of topics :

Complexity - O((NT)^T \* (N + T)^3)

N: count of words

T: number of topics

The algorithm is NP-hard for a large number of topics

<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.230.5953&rep=rep1&type=pdf> (assumed to be equal to T as our number of topics are small)

A simpler explanation of why the algorithm works-

<https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2>

<http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/>

LDA limitations:

* Fixed K (the number of topics is fixed and the count must be known ahead of time)
* Uncorrelated topics (Dirichlet topic distribution cannot capture correlations) (we use c\_v coherence score to overcome this)
* Non-hierarchical (in data-limited regimes hierarchical models allow sharing of data)
* Static (no evolution of topics over time)
* Bag of words (assumes words are exchangeable, sentence structure is not modeled)
* Unsupervised (sometimes weak supervision is desirable, e.g. in sentiment analysis)

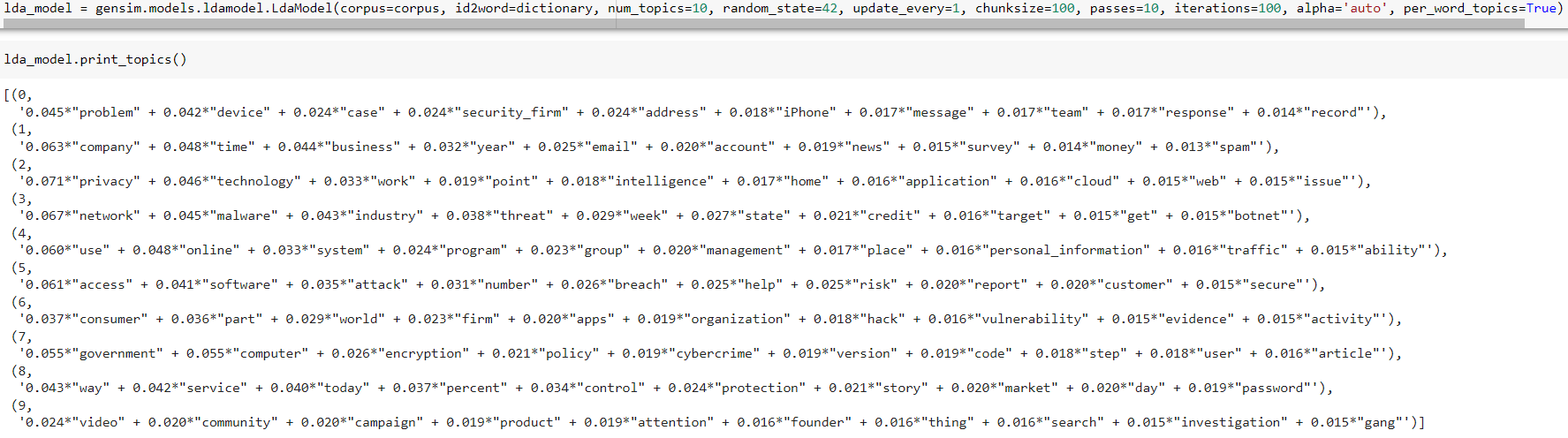
Paper referred:

* Tong and Zhang (2016): <https://www.airccj.org/CSCP/vol6/csit65316.pdf>
* Blei et al (2003) : <https://ai.stanford.edu/~ang/papers/jair03-lda.pdf>
* Spruit and Syed (2017) : <https://ieeexplore.ieee.org/document/8259775/authors#authors>
* Roy and Sontag (2011) : <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.230.5953&rep=rep1&type=pdf>

**C] Results and their inferences**

**I] Analysis on the non-peer-reviewed corpus: Post-Covid**

The print\_topics() method in the Gensim (Fig 1(a) ) library gives word-probability pairs for each topic. This tells us the probability that a word belongs to a particular topic. We can see that this value is only dependent on the frequency of a word under the topic. In Fig 1(a) we have chosen the num\_topics as 10 and the num\_words as 10. The num\_words parameter will print the top specified number of words with the highest probability for a topic. The value of num\_topics is initially randomly chosen for the base model.



Coherence score: 0.33738584980921493

Fig 1(a) Post covid non-peer-reviewed corpus- Base model

Now in order to improve the coherence value of our base model we need to find the best values for the hyperparameters. In the diagrams which follow for hyperparameter tuning, c is a label for coherence score.

1. **Hyperparameter tuning for chunksize**

Increasing the value of chunksize will reduce our training time as long as the chunk of documents easily fit in the memory. However as suggested by Hoffman et al, the value of chunksize also affects the quality of the model. Fig 1(b) shows how the coherence value changes with the value of chunksize. Based on this we choose the value of the peak which is chunksize=300 in our subsequent models for the non-peer reviewed corpus.

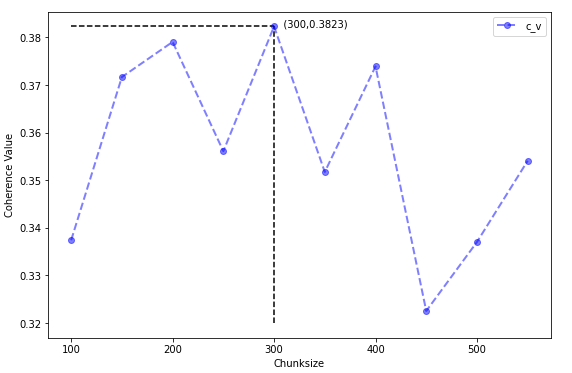


Fig 1(b) Plot between chunksize and coherence value, to obtain the best value for chunksize

1. **Hyperparameter tuning for passes:**

Passes control how often we train the model for the entire corpus. The value for the number of passes should be high enough so that the documents have converged. Fig 1(c) shows how the coherence value changes with the number of passes. Based on the peak we choose the value 40 as the number of passes in subsequent models for the non-peer-reviewed corpus.

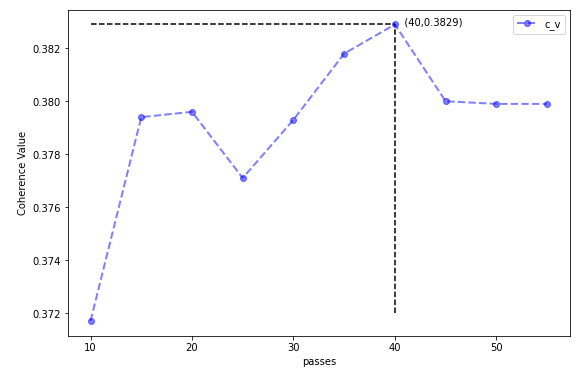


Fig 1(c) Plot between passes and coherence value, to obtain the best value of passes

1. **Hyperparameter tuning for iterations:**

Iterations control how often we repeat a particular loop over a document. A higher value of iterations is chosen so that the documents have converged. Fig 1(d) shows how the coherence value changes with the number of iterations for non-peer-reviewed corpus (Post-Covid). As we can see that all the documents have converged to a topic after 200 iterations hence there is no change in the coherence value. So the value of iterations is assigned as 200 for our subsequent models in non-peer-reviewed (Post-covid).

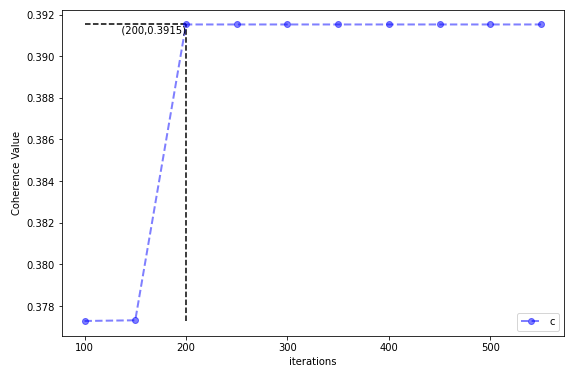


Fig 1(d) Plot between iterations and coherence value, to obtain the best value of iterations

1. **Hyperparameter tuning for the number of topics:**

Once the values of other hyperparameters are determined we can find the optimal number of topics by seeing how coherence value varies with it. Fig 1(e) shows how the coherence value changes with the number of topics. Based on this we can see that maximum coherence value is obtained at 16 topics. Hence in the final LDA model for Post-covid non-peer-reviewed corpus we set chunksize=300, passes=40, iterations=200, and number of topics = 16.

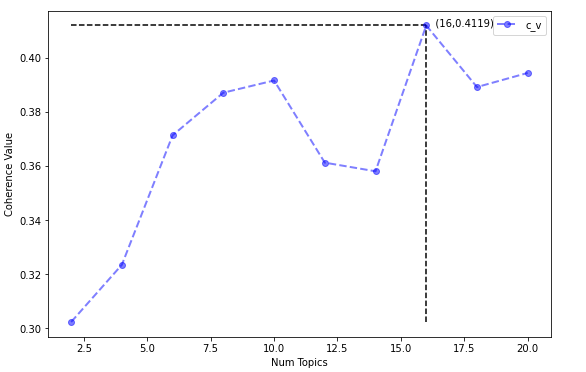


Fig 1(e) Plot between the number of topics and coherence value, to obtain the best value of the number of topics

Based on these values we obtain the following topics for the final model of Post-Covid non-peer-reviewed corpus.



Fig 1(f) Post-Covid non-peer-reviewed corpus- Final model

The following topics are inferred on the basis of the top 10 keywords for a topic.

Inferred topics for Post-Covid non-peer-reviewed corpus:

topic0: application vulnerability and flaws

topic1: cryptocurrency domain

topic2: agency investigations

topic3: botnet traffic attacks

topic4: firm fraud

topic5: software companies

topic6: computer insurance

topic7: cloud network

topic8: country's economy

topic9: financial fraud

topic10: account passcode hack

topic11: ransomware sites

topic12: encryption tools

topic13: phone technology

topic14: organized cybercrime

topic15: identity theft and privacy issues

After tuning the hyperparameters the coherence value for the post-covid non-peer reviewed corpus is 0.4119 which is a 22% improvement on the base model.

1. **pyLDAvis visualization for the final model:**

pyLDAvis is a web-based interactive visualization tool proposed by Sievert and Shirley, 2014. It is used to answer the following three questions

i) Meaning of each topic?

ii) How prevalent is each topic?

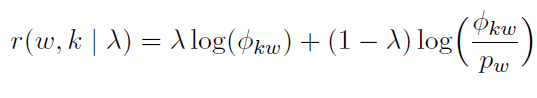
iii) How do the topics relate to each other?

The circles present in pyLDAvis plot represent topics obtained by the LDA model. The prevalence of a topic in the corpus is determined by the area of the circle. The intersection of circles with each other represents how related the two topics are. In pyLDAvis, the topics are numbered in decreasing order of prevalence.

Before we see how LDA can be used to answer (i), let us see how pyLDAvis produces this visualization. For example, let us consider that we want to obtain three topics from a corpus. We can represent these three topics as a triangle in a two-dimensional space, where each vertex represents a topic and the documents are represented as points. Based on how similar a document is to a topic it is placed close to that point. Similarly, we can extend this to four topics and a tetrahedron can be used to represent these topics in three dimensions. So to visualize K topics we need K-1 dimensions, since this is not possible for higher dimensions, Principal Component Analysis is used. pyLDAvis uses two principal components to visualize higher dimensions.

The meaning of each topic is obtained by how we interpret them. The weights assigned to keywords by the LDA model only give us information regarding the frequency of the term under the topic. This alone often does not suffice in interpreting topics. We also need to know how exclusive a word is to a topic. Bischof and Airoldi (2012) proposed to interpret a topic based on the frequency of the term under the topic as well as the term’s exclusivity to the topic. pyLDAvis uses this similar idea to define relevance.

The relevance of term w to topic k given as weight parameter 𝜆 (where 0≤ 𝜆 ≤ 1) as :



(Equation taken from Sievert C. and Shirley K (2014) https://nlp.stanford.edu/events/illvi2014/papers/sievert-illvi2014.pdf)

Here Φkw is the probability for term w under topic k. And pw is the marginal probability of term w in the corpus. So when 𝜆 =1 then the keywords are arranged only on the basis of the frequency of the term under the topic and when 𝜆 =0 then the keywords are arranged on the basis of its exclusivity (also known as lift, Taddy (2011)). It is important to note that changing the value of lambda in a pyLDAvis plot only helps us in interpreting the topics, it does not change the coherence value of the LDA model.

The bar graphs on the right of a pyLDAvis visualization tell us the frequency of a term in a topic (colored in red) with respect to its frequency in the entire corpus (colored in blue).

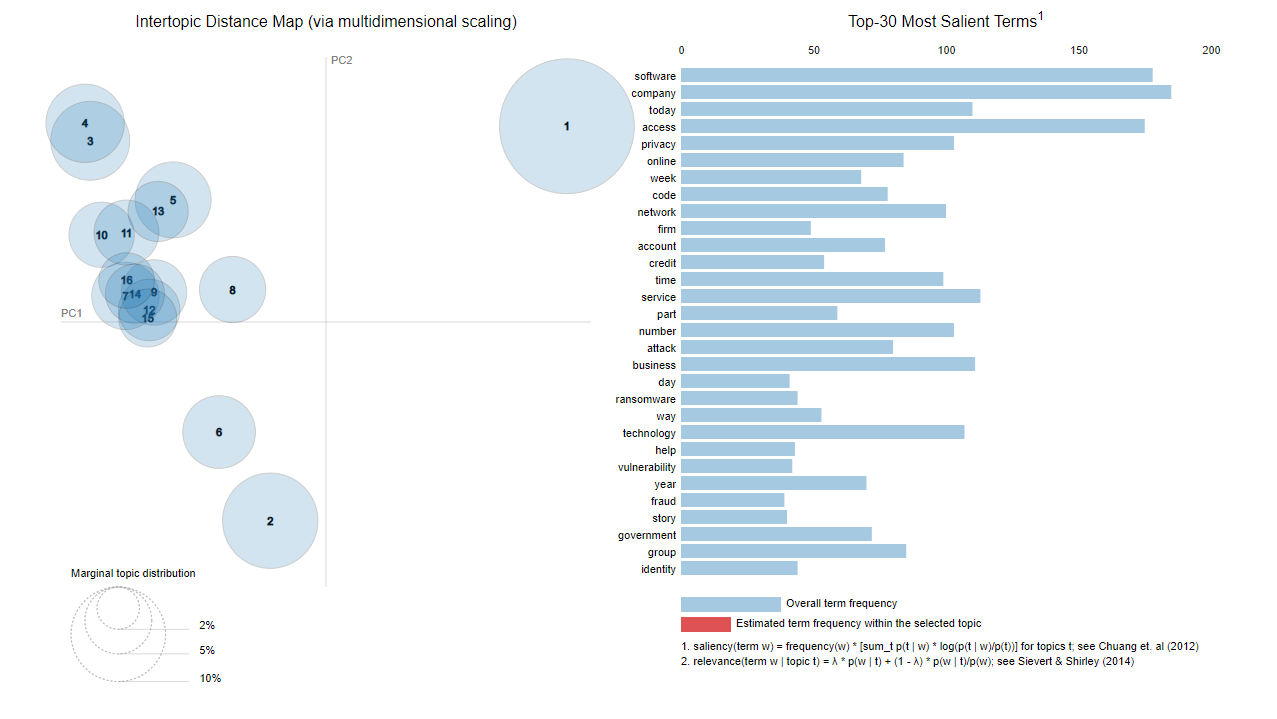


Fig 1(g) pyLDAvis plot for Post-Covid non-peer reviewed corpus- Final model

The blue bar graphs on the right hand side of Fig 1(g) represents the frequency of the top 30 keywords in the entire corpus. On selecting a particular topic (like topic 1 in Fig 1(h)) we get the frequency of that term in a topic (shown in red) with respect to its frequency on the entire corpus (shown in blue).

References:

1. Sievert C. and Shirley K (2014). *LDAvis: A method for visualizing and interpreting topics.* Accessed online: [Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces](https://nlp.stanford.edu/events/illvi2014/papers/sievert-illvi2014.pdf)
2. Jonathan M. Bischof and Edoardo M. Airoldi. 2012.Summarizing topical content with word frequency and exclusivity. ICML
3. Matthew A. Taddy 2011. On Estimation and Selection for Topic Models. AISTATS
4. "Online Learning for Latent Dirichlet Allocation", Hoffman et al. 2010.
5. Si Chen and Yufei Wang. Latent Dirichlet Allocation.

https://acsweb.ucsd.edu/~yuw176/report/lda.pdf

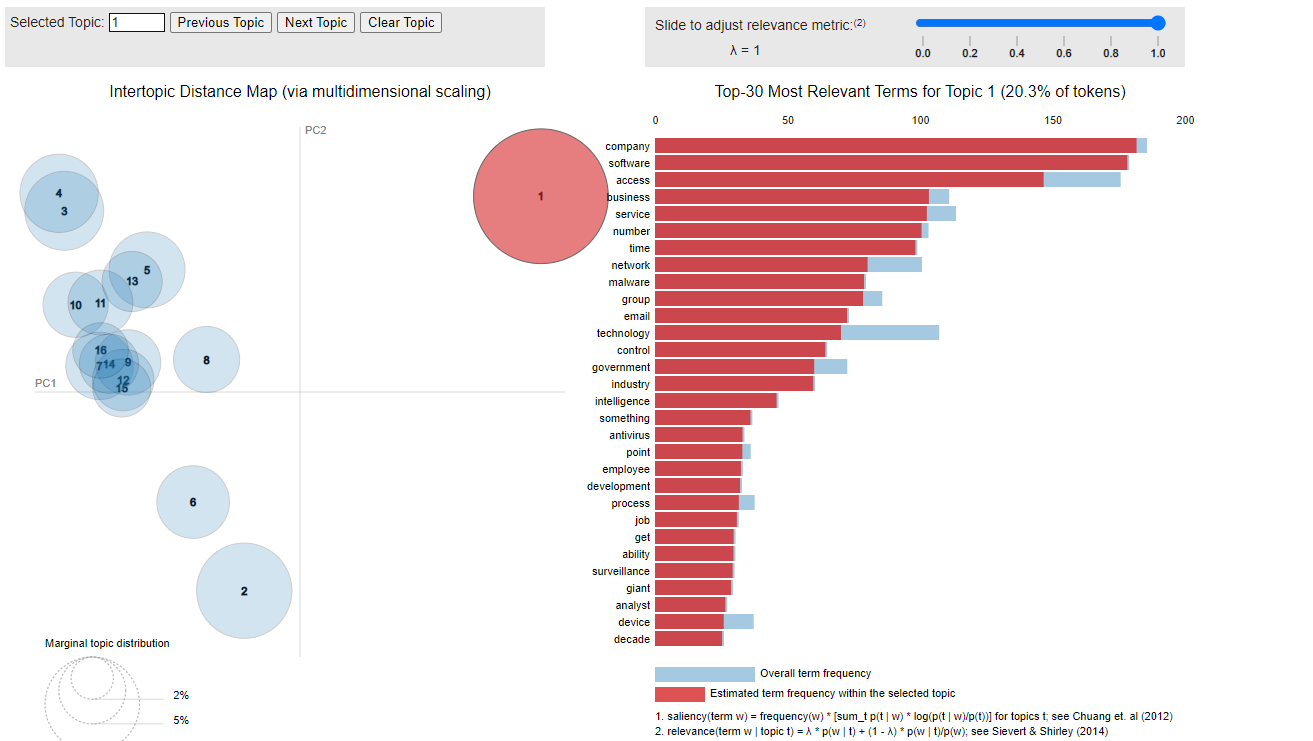


Fig 1(h) frequency of terms in topic 1 with respect to the entire corpus in post-Covid non-peer-reviewed corpus.

**f) Using word clouds to visualize topic weights**

The results in Fig 1(a) and Fig 1(f) can also be viewed using word clouds. Here the keywords in each topic are present in a cloud and the importance (weight assigned to a keyword) reflects its relative font size.

Fig 1(i) represents the topic cloud for the topics obtained from the final model of non-peer-reviewed post covid corpus.



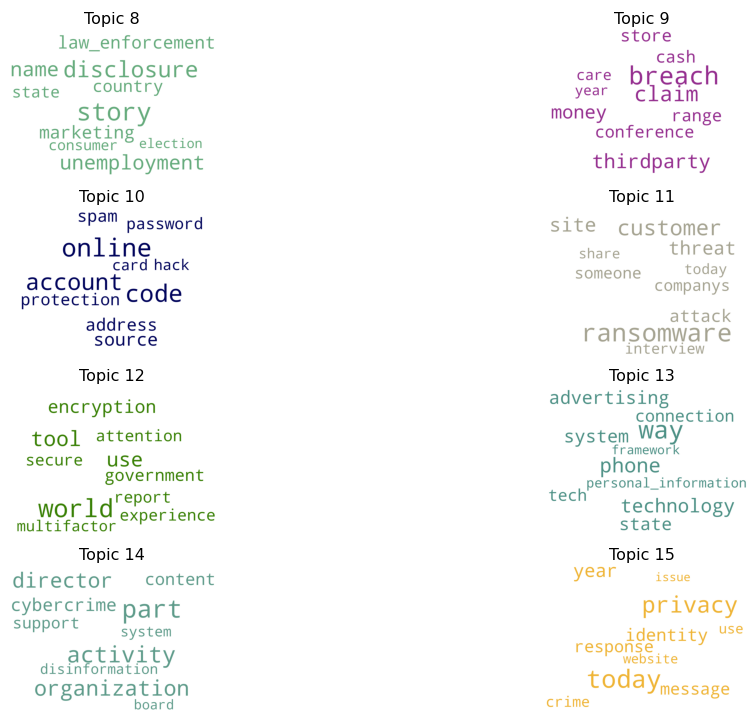


Fig 1(i) Word cloud for topics obtained from non-peer-reviewed post covid corpus

**II] Analysis on the non-peer-reviewed corpus: Pre-Covid**

A similar process of hyperparameter tuning as followed in section I is used for pre-covid non-peer-reviewed corpus. We obtain the following results for the final model.



Fig 2(a) Pre-covid non-peer-reviewed corpus - Final model

The following topics are inferred from the non-peer-reviewed corpus- Final model

Inferred topics:

topic0: customer data protection

topic1: Surveillance systems

topic2: system attacks

topic3: malware threats to businesses

topic4: database breaches

topic5: Periodic online reports

topic6: home services companies

topic7: network hacking

topic8: consumer insights

topic9: privacy protection

From Fig 1(f) and Fig 2(a) the following inferences are drawn, post covid we see an emergence of research in cryptocurrency, financial fraud, encryption tools, organized cybercrime, ransomware, cloud network security, etc.

Pre covid we saw topics like network hacking, database breaches, malware threats to businesses, system attacks, etc.

Common topics include hacking threats and privacy protection.

**III] Analysis on the peer-reviewed corpus: Pre-covid**

After applying hyperparameter tuning we obtain the following final model.



Fig 3(a) Pre-covid peer-reviewed corpus final model.

Topic inferred from pre-covid non-peer-reviewed corpus:

Topic0: industrial technology

Topic1: cyber privacy protection

Topic2: malware detection

Topic3: communication and power system

Topic4: Network intrusion

Topic5: Game theory for cyber security

Topic6: blockchain and cryptocurrency

Topic7: system attack model analysis

Topic8: computer design and development

Topic9: software cyber attack vulnerability

**IV] Analysis on the peer-reviewed corpus: Post-covid**

After applying hyperparameter tuning on the base model we obtain the following final model.



Fig 4(a) Post-covid peer-reviewed corpus final model.

Topic inferred:

Topic0: secure communication system

Topic1: privacy in healthcare

Topic2: software development and analysis

Topic3: Intelligence-based ransomware protection

Topic4: web awareness

Topic5: Breaches using code vulnerabilities

Topic6: power control system

Topic7: Data encryption

Topic8: malware solutions

Topic9: business risk management

Topic10: network security

Topic11: Insurance policy

Topic12: Digital Banking

Topic13: game model for honeypot

Topic14: optimized construction

From results I, II, III, and IV we can make the following table for coherence values.

|  | Pre-Covid | Post-Covid |
| --- | --- | --- |
| Peer reviewed | 0.44 | 0.46 |
| Non-peer reviewed | 0.40 | 0.4119 |

Fig 4(b) Coherence value table

In the analysis done till this point, we ran separate LDA models for pre and post covid periods. This process does not guarantee that the topics obtained will be the same, so instead, we can also run LDA on the entire corpus. Doing this we can see the distribution of papers for a topic in the pre and post-covid period.

**V] Analysis on peer-reviewed corpus**

After hyperparameter tuning for the papers from 2010-2021 we get the following final model.

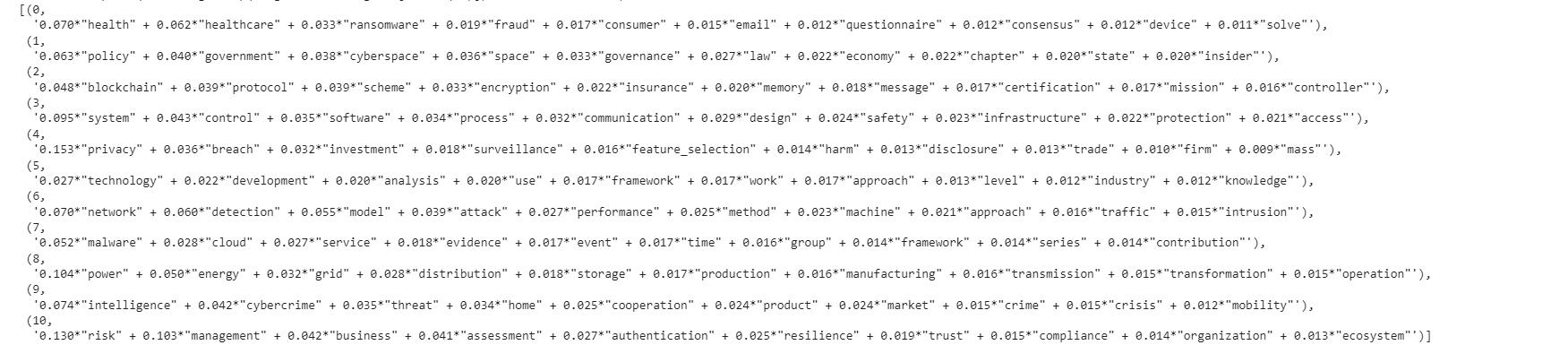


Fig 5(a) Peer-reviewed corpus- Final model

Coherence value: 0.44

Topic inferred from peer reviewed corpus:

Topic 0: Healthcare ransomware (Label : HR)

Topic 1: Government cyberspace policy (Label: GCP)

Topic 2: Blockchain encryption protocol ( Label: BP)

Topic 3: Control system safety (Label: CtrlS)

Topic 4: Privacy breach in trading firms (Label: PB)

Topic 5: Industry level technology development analysis ( Label: IT)

Topic 6: Network attack detection models (Label: NW)

Topic 7: Cloud service malware (Label: CSM)

Topic 8: Energy grid distribution (Label: EG)

Topic 9: Cybercrime threat intelligence (Label: CTI)

Topic 10: Risk management business authentication (Label: RBA)

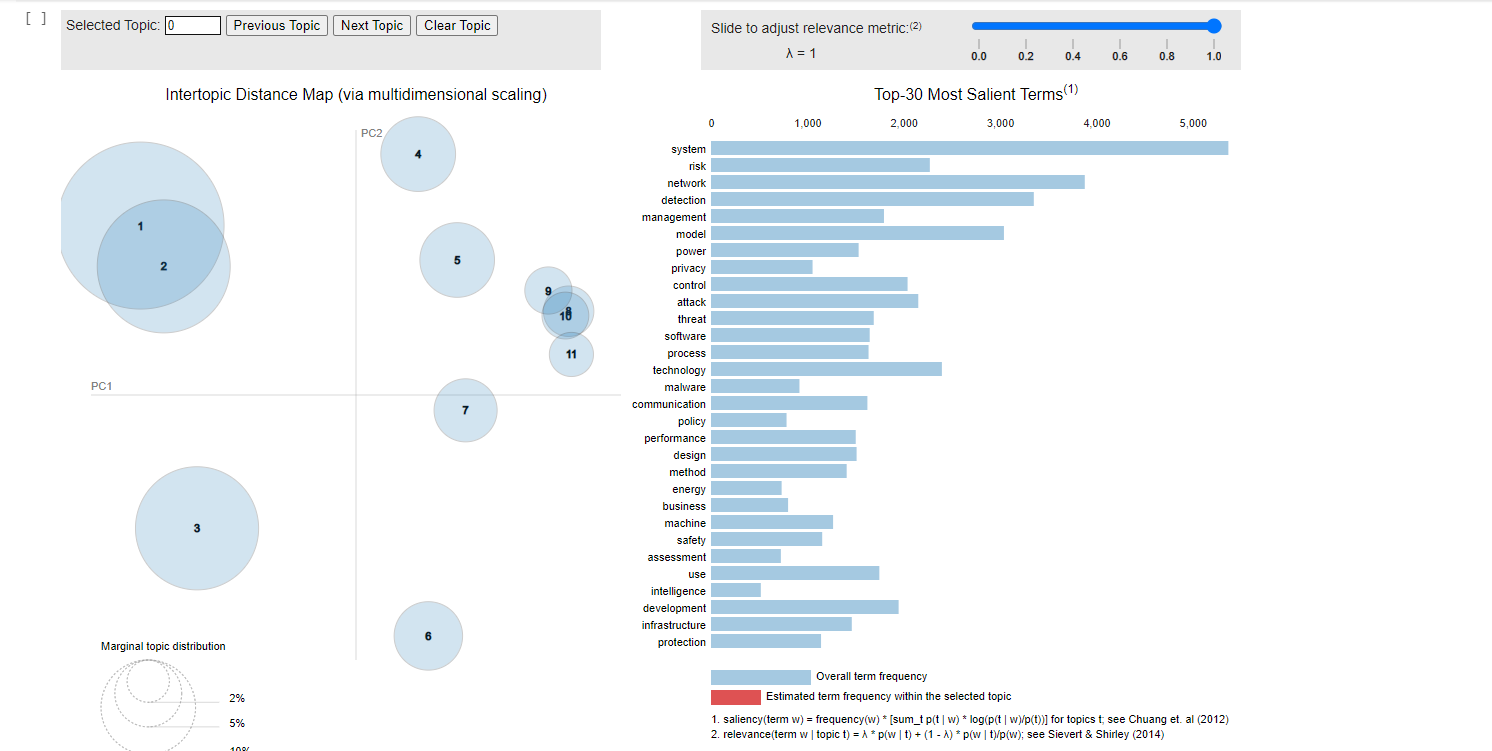


Fig 5(b) pyLDAvis plot for peer reviewed corpus-Final model

Each document in a corpus is assigned weights with respect to each document. In section I part e we have discussed how K topics can be represented in a K-1 dimensional space, and documents are placed close to one of the topics based on its similarity. We can use these weights to classify the documents based on the dominant topic.

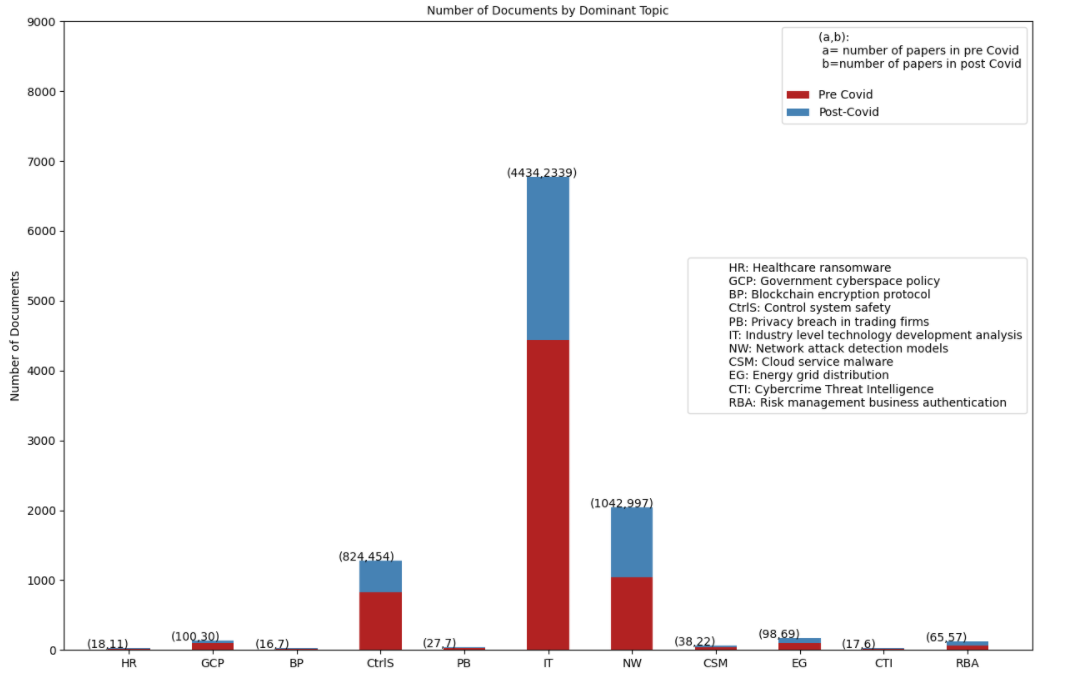


Fig 5(c) Stacked bar graph based on the dominant topic distribution of peer reviewed corpus

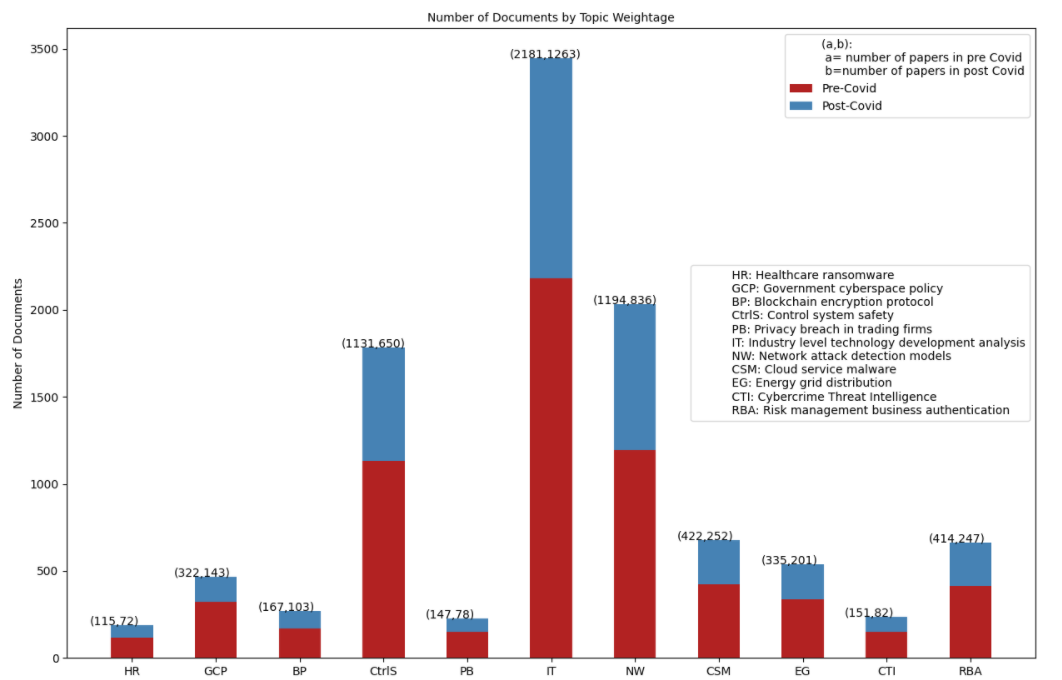


Fig 5(d) Stacked bar graph based on topic weightage distribution of peer-reviewed corpus

In Fig 5(c), a document is assigned to the topic which has the highest weightage. The y-axis represents the number of documents in a particular topic. Whereas the labels on the x-axis indicate the topic names of the peer-reviewed corpus. (Refer to section V Fig 5(b))

From Fig 5(c) we can infer that close to 40% percent of the peer-reviewed corpus is majorly related to Industry level technology development analysis. Whereas, the proportion of corpus which is majorly related to healthcare ransomware, blockchain encryption protocol, privacy breach in trading firms, cloud service malware, and cybercrime threat intelligence is almost negligible. We can also conclude that post covid, the number of papers majorly related to government cyberspace policy has significantly decreased.

On the other hand, in Fig 5(d) the number of documents for a topic is obtained by summing up the actual weight contributions of that topic in all documents. From Fig 5(d) we can conclude that most of the peer-reviewed papers are related to industry-level technology development analysis, network attack detection models, and control system safety in some way. We can also see that post covid, there has been a 7.92 % increase (66.37% of the post covid peer-reviewed corpus and 58.45% of the pre covid peer-reviewed corpus) in the percentage of papers related to industry level technology development analysis.

**Result VI- Year wise dominant topic distribution**

Now in order to visualize the distribution of papers in the pre and post covid era in an even better way we can show how the papers are distributed across the years for a given topic. The graphs shown below are a plot of the number of documents with years (from 2010 to 2021). For each topic we have shown a graph which shows the actual number of documents and a graph which shows the percentage of papers in a year related to that topic. Because of a lesser number of papers per year in the Pre-COVID period a percentage analysis can give a bias.

i) Topic 0: Healthcare ransomware (HR)

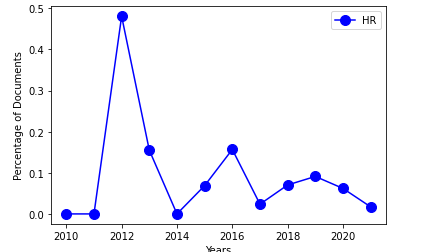


Fig 6(a) Analysis based on percentage

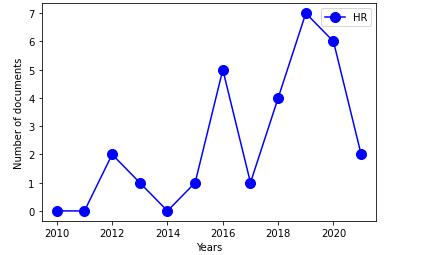


Fig 6(b) Analysis based on absolute number of papers

Inferences: From Fig 6(a) we can conclude that the percentage of papers related to healthcare ransomware is steadily decreasing post-covid.

From Fig 6(b) we can conclude that the number of papers related to healthcare ransomware is steadily decreasing post-covid.

ii) Topic 1: Government cyberspace policy (GCP)

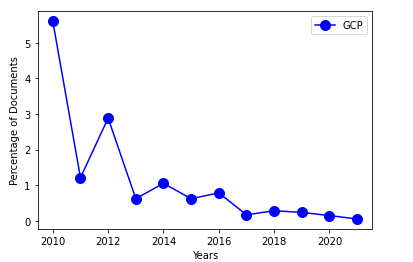


Fig 6(c) Percentage based analysis

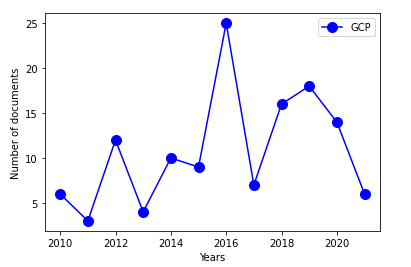


Fig 6(d) Analysis based on absolute number of papers

Inferences: From Fig 6(c) we can conclude that the percentage of papers related to government cyberspace policy is steadily decreasing post-covid.

From Fig 6(d) we can conclude that the number of papers related to government cyberspace policy is steadily decreasing post-covid.

iii) Topic 2: Blockchain encryption protocol (BP)

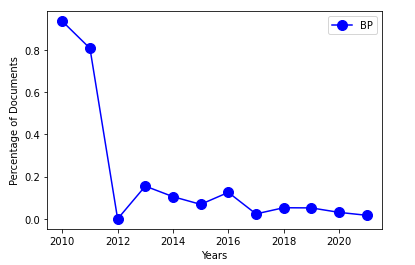


Fig 6(e) Percentage based analysis

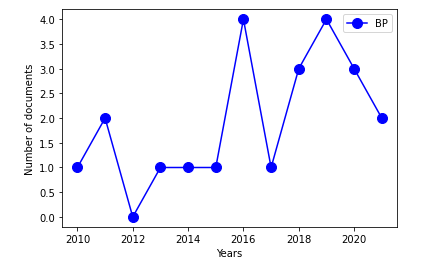


Fig 6(f) Analysis based on absolute number of papers

Inferences: From Fig 6(e) we can conclude that the percentage of papers related to blockchain encryption protocol is steadily decreasing post-covid.

From Fig 6(f) we can conclude that the number of papers related to blockchain encryption protocol is steadily decreasing post-covid. However, the absolute number is still higher than pre-covid years.

iv) Topic 3: Control system safety (CtrlS)

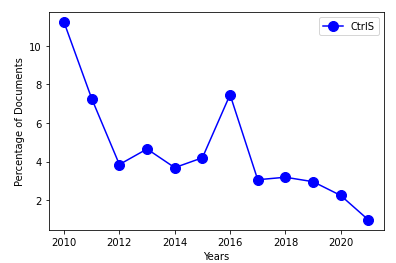


Fig 6(g) Percentage based analysis

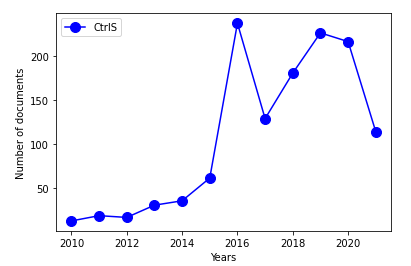


Fig 6(h) Analysis based on absolute number of papers

Inferences: From Fig 6(g) we can conclude that the percentage of papers related to control system safety is steadily decreasing post-covid.

From Fig 6(h) we can conclude that the number of papers related to control system safety is steadily decreasing post-covid.

v) Topic 4: Privacy breach in trading firms (PB)

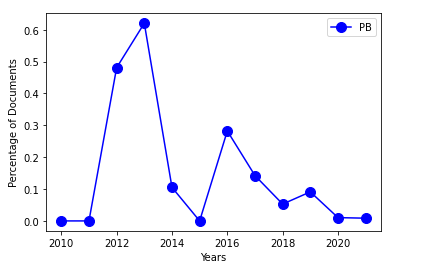


Fig 6(i) Percentage based analysis

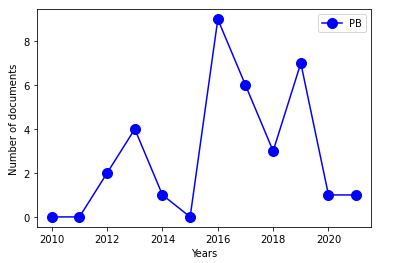


Fig 6(j) Analysis based on absolute number of papers

Inferences: From Fig 6(i) we can conclude that the percentage of papers related to privacy breach in trading firms is negligible and constant in the post-covid era.

From Fig 6(j) we can conclude that the number of papers related to privacy breach in trading firms is negligible in the post-covid era.

vi) Topic 5: Industry level technology development analysis (IT)

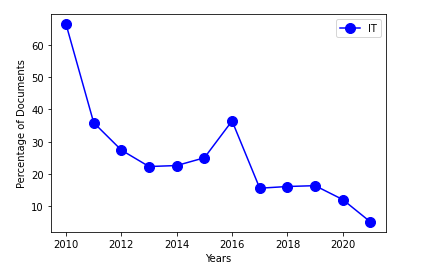


Fig 6(k) Percentage based analysis

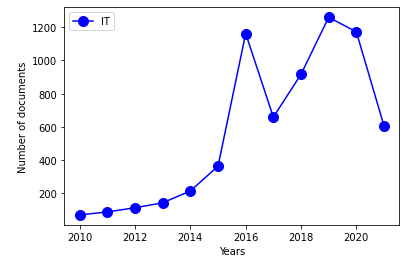


Fig 6(l) Analysis based on absolute number of papers

Inferences: From Fig 6(k) we can conclude that the percentage of papers related to industry level technology development analysis is steadily decreasing post-covid. From Fig 6(l) we can conclude that the number of papers related to industry level technology development analysis is steadily decreasing post-covid.

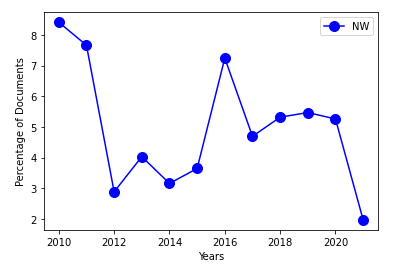
vii) Topic 6: Network attack detection models (NW)

Fig 6(m) Percentage based analysis

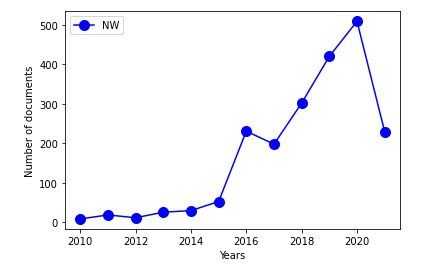


Fig 6(n) Analysis based on absolute number of papers

Inferences: From Fig 6(m) we can conclude that the percentage of papers related to network attack models has steadily decreased post-covid.

From Fig 6(n) we can conclude that the number of papers related to network attack models has increased then decreased post-covid.

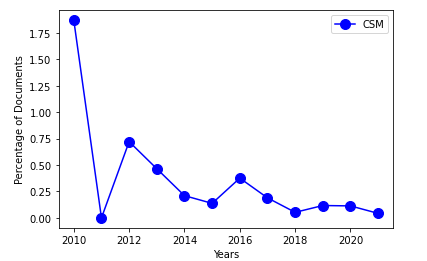
viii) Topic 7: Cloud service malwares (CSM)

Fig 6(o) Percentage based analysis

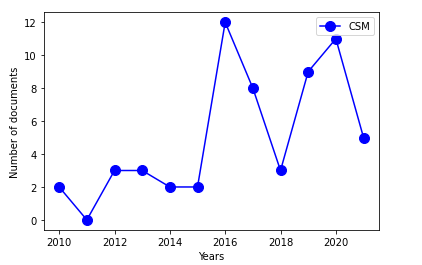


Fig 6(p) Analysis based on absolute number of papers

Inferences: From Fig 6(o) we can conclude that the percentage of papers related to cloud service malware is steadily decreasing post-covid.

From Fig 6(p) we can conclude that the number of papers related to cloud service malware has increased then decreased post-covid.

ix) Topic 8: Energy grid distribution (EG)

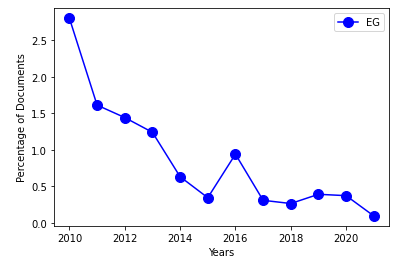


Fig 6(q) Percentage based analysis

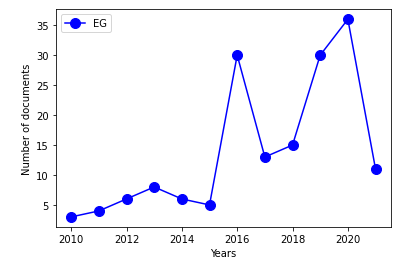


Fig 6(r) Analysis based on absolute number of papers

Inferences: From Fig 6(q) we can conclude that the percentage of papers related to energy grid distribution is steadily decreasing post-covid.

From Fig 6(r) we can conclude that the number of papers related to energy grid distribution has increased then decreased post-covid.

x) Topic 9: Cybercrime threat intelligence

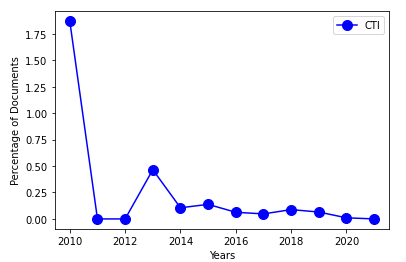


Fig 6(s) Percentage based analysis

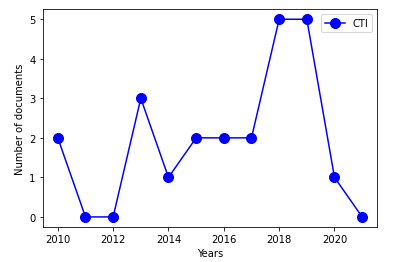


Fig 6(t) Analysis based on absolute number of papers

Inferences: From Fig 6(s) we can conclude that the percentage of papers related to cybercrime threat intelligence is steadily decreasing post-covid.

From Fig 6(t) we can conclude that the number of papers related to cybercrime threat intelligence is significantly decreasing post-covid.

xi) Topic 10: Risk management business authentication

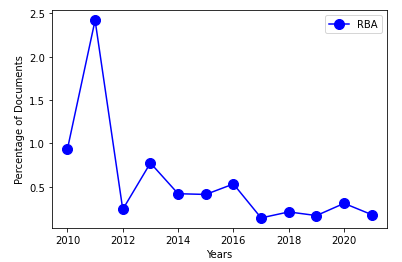


Fig 6(u) Percentage based analysis

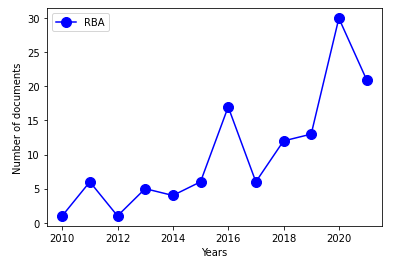


Fig 6(v) Analysis based on absolute number of papers

Inferences: From Fig 6(u) we can conclude that the percentage of papers related to risk management business authentication has increased then decreased post-covid. From Fig 6(v) we can conclude that the number of papers related to risk management business authentication increased significantly then decreased post-covid.

**Inferences:**

From the figures present in section VI we can finally conclude that in the Post-covid period we have seen a slight increase in the number of papers published in healthcare ransomware, control system safety, industry level technology development analysis, cloud service malwares, energy grid distribution, and risk management business authentication as compared to the pre-covid period. Whereas, during this period there is a decrease in the number of papers in government cyberspace, blockchain encryption protocol, and privacy breach in trading firms as compared to the pre-covid period.

**INFERENCE TABLE:**

| Figure Number | Inference(s) |
| --- | --- |
| 1(b) | chunksize=300, optimal value for post covid non-peer-reviewed corpus |
| 1(c) | passes=40, optimal value for post covid non-peer-reviewed corpus |
| 1(d) | iterations=200, optimal value for post covid non-peer-reviewed corpus. After 200 iterations the coherence value does not change with an increase in the number of iterations. This is because the documents have converged in 200 iterations for the relatively smaller corpus (1244 papers) |
| 1(e) | Number of topics=16 is the value with the highest coherence value. Hence, chunksize=300, passes=40, iterations=200, number of topics=16 are the set of hyperparameters to be used for the final model of the non-peer-reviewed post covid corpus. |
| 1(g) | The right hand side of the pyLDAvis plot shows us the top 30 words with highest frequency (blue bar graphs) in the non-peer-reviewed post covid corpus. |
| 1(h) | The right hand side shows us the top 30 words with highest frequency in topic 1 (red bar graphs). The blue bar graphs show the overall frequency in the entire non-peer-reviewed post covid corpus. |
| 1(i) | The size of the words in the word clouds for each topic show the importance of the word for that topic. |
| 1(f) and 2(a) | Post covid there is an emergence of research in cryptocurrency, financial fraud, encryption tools, organized cybercrime, ransomware, cloud network security, etc. Pre covid we saw topics like network hacking, database breaches, malware threats to businesses, system attacks, etc. Common topics include hacking threats and privacy protection. |
| 3(a) and 4(a) | Topics are inferred from the keywords for peer-reviewed pre covid corpus and peer-reviewed post covid corpus. (Please refer to the list below the respective figures) |
| 4(b) | Coherence values after hyperparameter tuning.   1. Non-peer-reviewed copus:  * Pre Covid: 0.40 * Post Covid: 0.4119  1. Peer-reviewed Corpus:  * Pre Covid: 0.44 * Post Covid: 0.46 |
| 5(a) | List of topics are inferred from the keywords along with their labels for peer-reviewed corpus (2010-2021). |
| 5(c) | Close to 40% percent of the peer-reviewed corpus is majorly related to Industry level technology development analysis. Whereas, the proportion of corpus which is majorly related to healthcare ransomware, blockchain encryption protocol, privacy breach in trading firms, cloud service malware, and cybercrime threat intelligence is almost negligible. We can also conclude that post covid, the number of papers majorly related to government cyberspace policy has significantly decreased. |
| 5(d) | Most of the peer-reviewed papers are related to industry-level technology development analysis, network attack detection models, and control system safety in some way. We can also see that post covid, there has been a 7.92 % increase (66.37% of the post covid peer-reviewed corpus and 58.45% of the pre covid peer-reviewed corpus) in the percentage of papers related to industry level technology development analysis. |
| 6(a) | We can conclude that the percentage of papers related to healthcare ransomware is steadily decreasing post-covid. |
| 6(b) | From Fig 6(b) we can conclude that the number of papers related to healthcare ransomware is steadily decreasing post-covid. |
| 6(c) | The percentage of papers related to government cyberspace policy is steadily decreasing post-covid. |
| 6(d) | The number of papers related to government cyberspace policy is steadily decreasing post-covid |
| 6(e) | The percentage of papers related to blockchain encryption protocol is steadily decreasing post-covid. |
| 6(f) | The number of papers related to blockchain encryption protocol is steadily decreasing post-covid. However, the absolute number is still higher than pre-covid years. |
| 6(g) | The percentage of papers related to control system safety is steadily decreasing post-covid. |
| 6(h) | The number of papers related to control system safety is steadily decreasing post-covid. |
| 6(i) | The percentage of papers related to privacy breach in trading firms is negligible and constant in the post-covid era. |
| 6(j) | The number of papers related to privacy breach in trading firms is negligible in the post-covid era. |
| 6(k) | The percentage of papers related to industry level technology development analysis is steadily decreasing post-covid. |
| 6(l) | The number of papers related to industry level technology development analysis is steadily decreasing post-covid. |
| 6(m) | The percentage of papers related to network attack models has steadily decreased post-covid. |
| 6(n) | The number of papers related to network attack models has increased then decreased post-covid. |
| 6(o) | The percentage of papers related to cloud service malware is steadily decreasing post-covid. |
| 6(p) | The number of papers related to cloud service malware has increased then decreased post-covid. |
| 6(q) | The percentage of papers related to energy grid distribution is steadily decreasing post-covid. |
| 6(r) | The number of papers related to energy grid distribution has increased then decreased post-covid. |
| 6(s) | The percentage of papers related to cybercrime threat intelligence is steadily decreasing post-covid. |
| 6(t) | The number of papers related to cybercrime threat intelligence is significantly decreasing post-covid. |
| 6(u) | The percentage of papers related to risk management business authentication has increased then decreased post-covid. |
| 6(v) | The number of papers related to risk management business authentication increased significantly then decreased post-covid. |

Related Work:

We have referred to Tong and Zhang (2016) to gain insight about how LDA is used in text mining and its benefits. The paper by Blei et al (2003) provided us a deeper understanding of the algorithm and the mathematics behind it. The reasons to choose c\_v as the quantitative measure to evaluate the interpretability of the topic model and its derivation is referred from Spruit and Syed (2017).

In order to determine the time complexity of LDA we referred to the work done by Roy and Sontag (2011). The work done by Hoffman et al (2010) helped us understand the impact of certain hyperparameters (like chunksize) on the performance of the LDA model. The work done by Chen and Wang helped us visualize how the documents converge in an LDA model after subsequent iterations. The concept of exclusivity of a keyword in a topic explained by Taddy (2011) helped us understand how word frequency isn’t the only way to assign importance to a keyword in a topic. Subsequently, the idea to use exclusivity and word frequency was provided by Bischof and Airoldi (2012).

We referred to Sievert and Shirley (2014) to understand the tool used to visualize and interpret topic models, pyLDAvis. This paper also helped us understand how a parameter λ is used to assign priority to either word frequency or exclusivity in the relevance metric used by pyLDAvis and how PCA is used to create 2D plots.

Links to papers:

* Tong and Zhang (2016): <https://www.airccj.org/CSCP/vol6/csit65316.pdf>
* Blei et al (2003) : <https://ai.stanford.edu/~ang/papers/jair03-lda.pdf>
* Spruit and Syed (2017) : <https://ieeexplore.ieee.org/document/8259775/authors#authors>
* Roy and Sontag (2011) : <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.230.5953&rep=rep1&type=pdf>
* Hoffman et al (2010): <https://papers.nips.cc/paper/2010/file/71f6278d140af599e06ad9bf1ba03cb0-Paper.pdf>
* Chen and Wang : <https://acsweb.ucsd.edu/~yuw176/report/lda.pdf>
* Taddy (2011) : <http://proceedings.mlr.press/v22/taddy12/taddy12.pdf>
* Bischof and Airoldi (2012) : <https://icml.cc/Conferences/2012/papers/113.pdf>
* Sievert and Shirley (2014) : <https://nlp.stanford.edu/events/illvi2014/papers/sievert-illvi2014.pdf>

Other papers referred -

<https://drive.google.com/drive/folders/14wnpGZGPrvDdc9JRwiel_zp2J0zUjR62?usp=sharing>

**Future Work**

**Things that we weren’t able to do :**

* Running our analysis on a larger data-set acquired from different research paper banks.

Reason: The large amount of time taken to tune each parameter of the model because of the time complexity of the algorithm (NP hard algorithm for a large corpus)

* Tuning our hyperparameters in the most optimal way - we lineary tuned the 4 parameters in one order only, there are 24 different permutations in which this could have been done. Also, the most optimal way of tuning these parameters would be using 4 for loops for each parameter.

Reason: The large amount of time taken to tune each parameter of the model because of the time complexity of the algorithm (NP hard algorithm for a large corpus)

* Using other measures to evaluate the model and compare the results obtained.

Reason: We only chose and used the most popular measure because it had been well researched upon.

* To find a better/efficient way to automatically interpret topics (we interpreted all topics manually)

Reason: We found some methods to achieve this, but they didn’t give satisfactory results