ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Paper ID #44920

# Statistical Word Analysis to support the Semiautomatic Implementation of the NIST 800-53 Cybersecurity Framework

**Dr. Mirco Speretta, Fairfield University**

Rohan Sahu is a senior at Westhill High School in Stamford, Connecticut. He started to learn about statistical word analysis based on TF-IDF in the fall of 2021, when he was a sophomore. He implemented this technique from scratch in Java and applied it to the NIST Risk Management framework.

Dr. Mirco Speretta is the Director of the Cybersecurity Programs at Fairfield University. Before this role he spent 10 years as a director of technical engineering, acting as a security incident manager for mobile websites and applications. His early research focused on the development of semiautomatic techniques to build ontologies and the creation of user profiles that improve search results.

# Statistical Word Analysis to support the Semiautomatic Implementation of the NIST 800-53 Cybersecurity Framework

*Rohan Sahu[1] and Mirco Speretta[2]*

[1]Westhill High *School, Stamford, CT 06902; [2]*School of Engineering and Computing, Fairfield University, Fairfield, CT 06824

**Abstract:**

Cybersecurity frameworks such as NIST, CIS, and ISO, include a collection of families and controls that recommend security policies to organizations. They play a critical role in mitigating the risks of cyber attacks and breaches in organizations. Due to the manual process of selecting families and controls, the implementation of these frameworks is very resource-intensive and time-consuming. This project addresses this challenge by investigating the feasibility of partially automating the process of selecting families. In this study, we developed an application in Java that applies statistical techniques such as TF-IDF and Cosine similarity to the families of the NIST cybersecurity framework. The framework is split into a distinctive corpora of tokens representing each family. A corpus includes all the controls for a given family and is simplified to the list of tokens that are most representative of that family. We evaluated how accurately the corpora represented the framework by using both a qualitative and a quantitative approach. Considering the positive results of our tests, we believe that this approach could have a great impact on semi-automating the process of selecting controls within a family. This will reduce the resources and the cost needed for implementing cybersecurity frameworks. At the same time, it will increase the accuracy and consistency of the selection process.

## Introduction:

The use of computers and other personal electronic devices has grown at an exponential rate and has made a significant impact on our lives. Their staggering growth and development has also brought along a host of risks and security concerns as well. Today more than ever, sensitive data such as credit card information, address, and social security number is being shared over the internet through these devices and as a result the threat of cyber crime has dramatically increased. In fact, there are around 2,200 new cyberattacks everyday which equates to around 800,000 cyberattacks per year[1]. These cyberattacks come in a variety of forms including: Ransomware attack, phishing attack, and Distributed Denial of Service (DDoS) attack. The plethora of cyber attacks can target individuals as well as multinational corporations. Last year, over 6.9 billion dollars was lost due to cybercrime in America with over 850,000 different incidents reported[2]. One of the most notable incidents of cyber crime in recent years was the SolarWinds cyberattack in 2020. A group of foreign hackers were able to breach into the systems of the Texas-based tech company, SolarWinds, and cause over 18,000 customers to accidentally download a virus which infected their computers[3]. The severity and damage of this incident as well as many others have created a surge in the necessity for companies and agencies

to start implementing policies that can help mitigate these attacks. The mitigation of such incidents begins from the employees themselves. Around 95% of cybersecurity breaches stem from human error[4]. With this in mind, the Department of Defense as well as other organizations have worked to create a cybersecurity framework, a recommended set of guidelines and protocols that companies should comply with in order to ensure that they are using the best practices against cyberattacks[5]. Though there are a variety of cybersecurity organizations that provide frameworks including the National Institute of Standards and Technology (NIST), the International Organization of Standardization(ISO), and the Center for Internet Security (CIS). This project focuses on the NIST Risk Management framework.

The major problem companies face currently is that the process of fully implementing these frameworks can be very expensive and time-consuming. Though around 84% of organizations have instituted at least one of the cybersecurity frameworks, 64% of these organizations have only partially implemented them due to their high cost[8]. This leaves a large portion of companies in America unprotected against all forms of cyber threats, potentially resulting in huge financial loss. Every organization is unique and as a result requires a personalized cybersecurity plan. Companies strive to only implement families and controls which are applicable for them, which in most cases is not the entire NIST framework. 95% of organizations face significant challenges when implementing leading cybersecurity frameworks[9]. To be fully compliant with NIST's cybersecurity standards, it can take up to several years[10] and can range from $40,000 to $140,000 depending on the company's existing security and manpower[11]. One solution to this problem is to make the implementation process less manual. Despite many organizations clamoring for the automation of this process, there are no efficient methods currently available to the public and they continue to manually complete the process. Though applications such as *CyberArrow*[12] attempt to automate the process, they have not been able to achieve enough accuracy to be implemented by many organizations. In this study we aim to address this gap by applying multiple numerical statistical techniques and algorithms on the NIST 800-53 Cybersecurity Framework to test the feasibility of implementation in it using a semi-automatic approach.

## Background and Methodology:

NIST[6] identifies several different ways that organizations can use the framework. The Framework is made up of three components: the Framework Core, Profiles, and Tiers. Organizations can use the Framework to compare their current cybersecurity activities with those outlined in the Core to find out which areas they are achieving the outcomes described in the Core and which areas they may want to improve. The Framework lists steps that an organization can follow (such as creating a Current Profile and creating a Target Profile) to use the Framework to create a new cybersecurity program or to improve an existing one.
Because the Framework establishes a common language to communicate cybersecurity requirements, an organization can use the Framework to implement the cybersecurity requirements. Organizations can use these three components together to conduct a comprehensive review of their cybersecurity program. The framework is intended to serve 5 functions or activities: Identify, Protect, Detect, Respond, and Recover. Each of these functions are divided into categories and subcategories of cybersecurity activities and outcomes. These categories and subcategories then point to specific industry-accepted standards and guidelines (e.g., COBIT 5, ISO 27001) that provide more in-depth instruction on how to achieve each

specific activity or outcome. A collection of related categories or subcategories constructs a family. Families often focus on one aspect of an organization such as Asset Management, which deals with how an organization should protect its assets, or Business Environment, which centers around what a safe business environment looks like. NIST 800-53[7], the specific version of the framework we used, contains 20 different families spanning from Incident Control to Response Planning. Each of these families has a plethora of controls within them which instruct the organization on what practices they should implement. NIST 800-53 includes more than 1,000 controls.

*Review the Implementation guidelines laid out in the policies that form each of the available versions of NIST and select the version that would be most appropriate for the study:*

We reviewed the two main versions of NIST: NIST 800-53 and NIST 800-171 and, for this study, we decided to use the first one: it provides a richer sets of controls (i.e. more than 1,000 vs more than 100) that are organized in 21 families and it covers organizations from a broader pool.

*Development of the code using the TF-IDF algorithm to compile the top 10 most representative words for each family in NIST 800-53 framework:*

Once the framework was chosen, we needed to distinguish the families listed above by their most relevant and representative words. The words found in the documentation for each family will be referred to as *tokens*, and the set of ranked tokens that represent each family make up a *corpus*. The goal is to create 21 separate corpora for each family in the framework. The specific algorithm we used to build the corpora was the Text Frequency-Inverse Document Frequency (TF-IDF) algorithm[13]. This algorithm is very popular for information retrieval and word analysis, in fact it is one of the main techniques employed in Google's search engine[14]. Its ability to filter out the most relevant subset of words from a large dataset lends itself perfectly to the idea of establishing a list of representative tokens for each corpus in the NIST framework. Tokens which have more weight, or greater TF-IDF value are considered more relevant than tokens that have a lower value. The manner in which it is calculated is represented by the formula shown in Equation 1A:[15].

**Equation 1A: TF-IDF Formula**

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = log\frac{N}{1+df}$$

$$TF-IDF(t,d) = TF(t,d) * IDF(t)$$

This equation represents tf-idf as a function of terms and documents, represented by t and d, respectively. Term frequency (TF) is equal to the number of times a word appears in a document divided by the total number of words in the document, tf(t,d). Inverse document frequency (IDF) is equivalent to log(N/(df + 1)), where N represents the total number of documents and df represents document frequency which is the number of documents in which a word appears in. 1

is added to the document frequency, in case the frequency is equal to 0. I took the logarithm of the quotient because it normalizes and dampens the effect IDF has in the equation. TF-IDF is equal to the product of term frequency and inverse document frequency.

We developed the code for the algorithm in Java to calculate the TF-IDF weightage for each token. In order to ensure the accuracy and relevance of the results, all stop words (popular words that don't add any information to the text such as and, or, because) were discounted from the algorithm. The list of stopwords we used was obtained from *Ranks NL*[16] and it contains 180 words. Additionally, all the words were converted into lowercase and all non-alphanumeric symbols were removed, to avoid overcounting words. The code for the method is shown below:

*Using the TF-IDF Algorithm to Compile the top 10 most representative words for each family using my program:*

After developing the code for the TF-IDF algorithm, we ran the Java program over each corpus (i.e. each family in the framework.) This generated a list of tokens ranked by the TF-IDF weight. The token along with its TF-IDF value are then stored in a map as a key and value, respectively. Then the code parses through the map and sorts all the tokens based on decreasing TF-IDF value. The ten tokens with the greatest TF-IDF value made up the most representative words for each family. An example can be found in Table 1.

*Qualitative Validation of the Results from my Algorithm: Gather information via interviews of key stakeholders and compare the results to my algorithm*

To validate our results, we compared the top 10 words from each family provided by my algorithm with the NIST standards from an organization. I worked with Kari-Out Company, a New York based manufacturing company, that specializes in producing to-go food packaging, condiments, and other food products for quick service restaurants[17]. I conducted interviews with the Chief Information Security Officer (CISO), Chief Information Officer (CIO), Chief Executive Office (CEO), Chief HR officer (CHRO), Chief Operating Officer (COO) to gauge their security needs. Based on the responses we gleaned out the keywords that emphasized their security requirements and compared those keywords to the tokens produced by our algorithm.

For example, one question that was asked to all the employees during the interviews was, "What are the greatest risks to this organization?" This question is designed to help determine how well the organization complies to the NIST standards found under the *Risk Management* family. We recorded the most important concepts and ideas that were addressed by the employees and compared them to the most representative tokens that my algorithm produced for the Risk Management family. If the results from the interview and algorithm contained identical concepts, it would validate our algorithm and prove that the tokens are representative of the corpora. Additionally, if some of the tokens from our program were not discussed adequately during the interview, then controls which address those tokens should be implemented. Similarly, if the employees mentioned an idea repeatedly during the interview which isn't represented in the tokens, it would provide an area of improvement for our program. The results from this comparison as well as a specific example can be found in Figure 1.

*Automation of the Implementation Process: Using Cosine Similarity Analysis to Automatically Select Families to be Implemented by Kari-Out*

After developing a distinctive corpus containing representative tokens for each family, we decided to test whether the program was able to automatically classify security documents from *Kari-out* to a family in the framework. We compiled a list of the top 100 tokens in each corpus and stored them into a document. Next, we used the cosine similarity algorithm[18] which returns a value between 0-1, signifying how similar the document containing the most representative tokens for a family is to the security document provided by *Kari-out*. Then, we found the top 3 families whose corpora had the highest cosine similarity to the given security document. These results represent the families that our algorithm suggests that the company should implement. We compared the suggestions from our algorithm to the actual decisions made by the company on which family to implement based on the same document. Identical results would signify that the algorithm is successful in classifying documents into their appropriate families in the framework. Results from using the cosine similarity algorithm can be found in Table 3.

The formula for cosine similarity is described in Equation 1B[18]:

**Equation 1B: Cosine Similarity Formula**

$$\text{cosine}(\mathbf{v},\mathbf{w}) = \frac{\mathbf{v}\cdot\mathbf{w}}{|\mathbf{v}||\mathbf{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

**Results:**

Table 2 displays the top ten words with the highest TF-IDF values from the Asset Management family. Words that are asterisked represent words that are also found in the most representative words of other families. For example, *strategy* is also included within the top 10 most representative words of the *Data Security* family. The overlap in representative words between different families is further described in Table 2. The ranked words for each family in the framework can be found in the *Appendix*.

**Table 1: Compilation of the top 10 most representative words for each family using our program**

| Asset Management | |
|---|---|
| Word | TF-IDF Value |
| *organization | 7.92 |
| *management | 7.85 |

| | |
|---|---|
| *strategy | 7.21 |
| asset | 6.95 |
| *risk | 6.76 |
| *data | 6.71 |
| business | 6.35 |
| devices | 5.81 |
| achieve | 4.39 |
| importance | 4.01 |

Table 2 displays a chart representing how distinctive the corpora are from each other when the number of representative words are varied. The shade of green becomes lighter as the number of families in which there are overlapping tokens increases, representing a lower level of distinction among corpora in the framework. Though the use of TF-IDF was largely successful in this project, there are some limitations to it as well. The results shown in Table 2 show how as the length of the list is increased, the tokens begin to overlap between a greater number of families and the level of distinction between corpora decreases. The less distinction there is between corpora, the less accurate the algorithm will be in suggesting ideas or families that should be implemented. Thus the program is most effective when a smaller set of tokens are used to represent each corpus, and can become less accurate as the representative set of tokens grows in size.

**Table 2: Distinctiveness between corpora**

Key: xF represents the number of tokens that overlap between x families

| **Total Tokens** | **Unique Tokens** | **2F** | **3F** | **4F** | **5F** | **6F** | **7F** | **8F** | **9F** | **10F** | **11F** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **105 (top 5)** | **80** | **17** | **5** | **3** | **0** | | | | | | |
| **210 (top 10)** | **143** | **39** | **13** | **6** | **3** | **1** | **1** | **0** | | | |
| **420 (top 20)** | **241** | **82** | **40** | **26** | **18** | **7** | **4** | **2** | **1** | **1** | **0** |

*Results of Qualitative Validation of the algorithm: Comparison of key ideas from interviews with the suggestions from the TF-IDF Algorithm*

Figure 1 is a Venn-diagram representation of the suggestions our algorithm provided for the implementation of the Risk Management family compared with the key ideas and responses recorded from the interviews when employees of *Kari-out* were asked about managing risks. The suggestions from our algorithm originate from the list of the most representative tokens for the Risk Management family produced by the TF-IDF algorithm. The intersection represents ideas and concepts that were reflected in both our algorithm and the interviews for the Risk Management Family. The numbers next to the tokens found in the intersection represent the rank

of each representative token of the family. The average, 4.57, shows that the top *4.57* tokens from this family are represented in both the interviews and our algorithm. In other words, the most relevant tokens to the Risk Management family are common in both our algorithm and the interviews.

**Risk Management**

**Figure 1: Comparison between results from algorithm and data from Kari-Out**

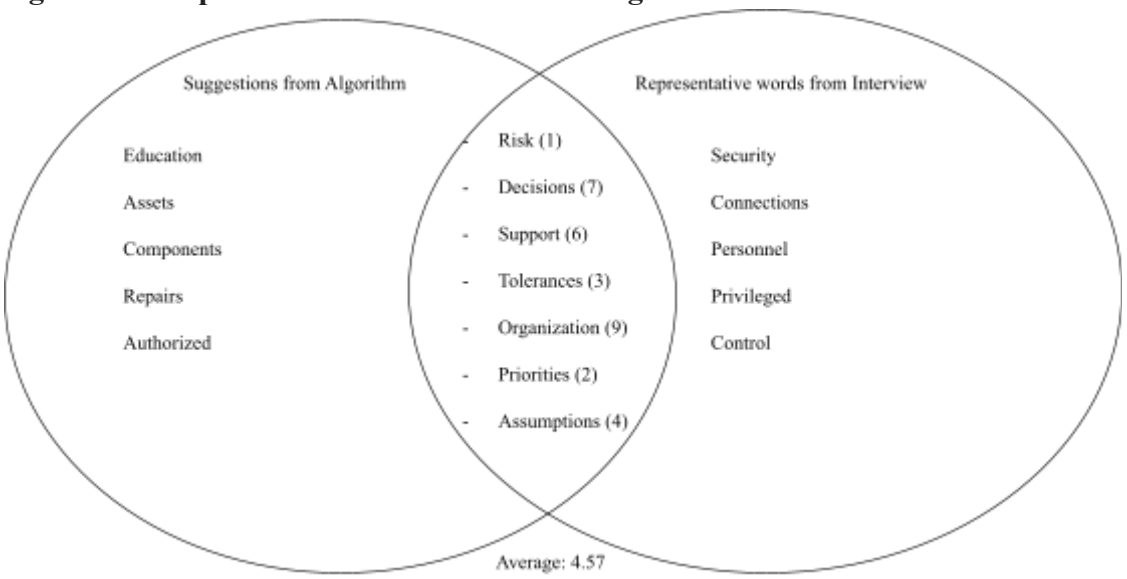
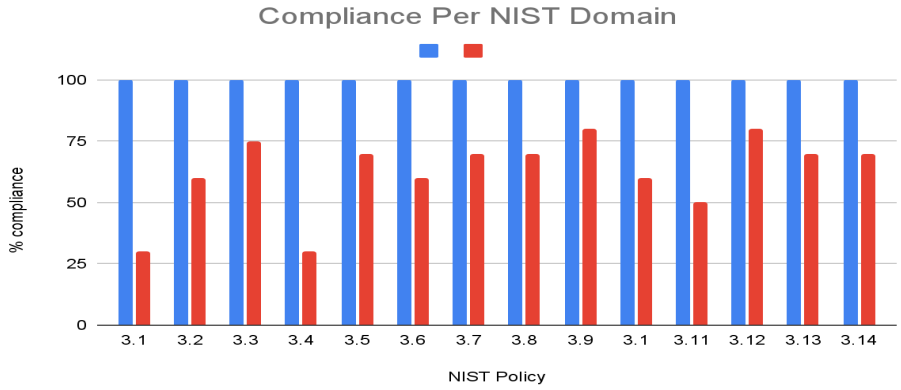
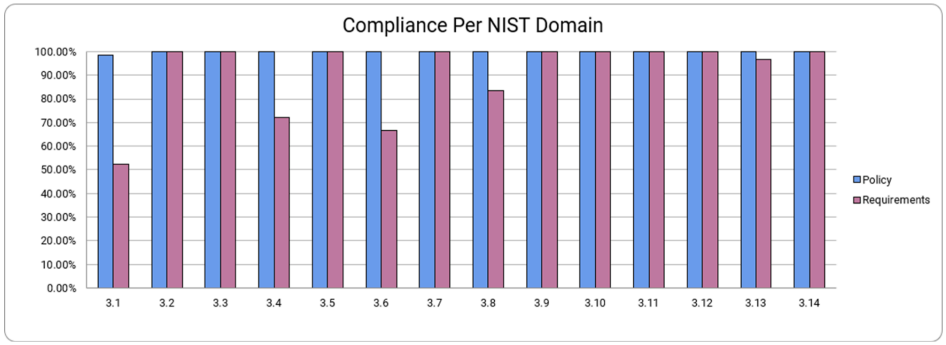



*Figure 2* displays the progress *Kari-out* made in its compliance with the NIST framework after using the suggestions from our algorithm described in *Figure 1*, for the controls found in the Risk Management Family. Compliance is measured in *Kari-Out* by the accumulation of documentation and the enforcement of policies that address a specific control.

**Figure 2: Drastic Improvement in the NIST Implementation process at Kari-out using the suggestions provided by our program (Pre and Post)**




**Pre**
Blue: Required compliance
Red: Current Compliance




**Post**

*Results from Automating the Selection of Families Using Cosine Similarity on Kari-out's Security Documents*

Table 3 displays the cosine similarity values between the top 100 most representative tokens from each corpus in the framework and a security document from Kari-out. The three families which Kari-Out chose to implement more controls based on the security document are highlighted. As shown in Table 3, the Cosine Similarity algorithm was able to successfully recommend *Kari-Out* to implement controls from an appropriate family when given a security document. The actual decision made by Kari-Out is also within the top three suggestions our algorithm made based on the same security document. This shows that our program can automatically classify a security-related document into a family in the NIST framework with high accuracy, which could dramatically aid companies in their NIST implementation process.

**Table 3: Cosine Similarity Results**

**Firewall Document**

| Rank | Family | Cosine Similarity |
|---|---|---|
| 1 | Information Protection Processes | 0.4486632382 |
| 2 | Data Security | 0.4339489467 |
| 3 | Protective Technology | 0.3864591173 |

**Workstation Security Document**

| Rank | Family | Cosine Similarity |
|---|---|---|
| 1 | Protective Technology | 0.3843075691 |
| 2 | Business Environment | 0.3679464844 |
| 3 | Security Continuous Monitoring | 0.3508232077 |

**Personnel Security Procedures**

| Rank | Family | Cosine Similarity |
|---|---|---|
| 1 | Information Protection Processes | 0.5085476277 |
| 2 | Risk Assessment | 0.4688577462 |
| 3 | Asset Management | 0.4203857014 |

## Conclusions and Future Work:

The NIST 800-53 cybersecurity framework represents a big challenge for all the organizations that need to implement it because of the time consuming and manual effort. In this project, we proposed and investigated the application of well known statistical and word analysis techniques to NIST 800-53. We believe that our approach can support the semi-automatic implementation of the framework. We validated the proposed technique in various ways, including a real world application: an organization that was undergoing the process of implementing NIST. Our approach proved to be successful and beneficial for the company.

The improvement in compliance sparked by the recommendations provided by our program show how powerful and beneficial it can be for companies. Kari-Out was able to instantly achieve higher results of compliance, which would have otherwise taken a substantially longer period of time and required more resources. While the results from this study are based on sample data we believe that this approach has the potential to provide suggestions for what families an organization should implement given data from interviews or security documents. Based on these positive results we believe that our approach can be further investigated in various ways. First, these preliminary results should be tested over more data to validate statistical significance. We should consider taking into accounts concepts that may not be present within the list of representative tokens for each corpus in the framework. Also we should look into the semi-automatic selections of controls within each family.

## References


1. NortonOnline. (n.d.). *115 cybersecurity statistics + trends to know in 2023*. Official Site. Retrieved March 2, 2023, from https://us.norton.com/internetsecurity-emerging-threats-cybersecurity-statistics.html#

2. Skiba, K. (2022, November 30). *FBI: Nearly $7 billion lost to cybercrime in 2021*. AARP. Retrieved March 2, 2023, from https://www.aarp.org/money/scams-fraud/info-2022/fbi-internet-crime-report.html#:~:text=Americans%20was%20hit%20by%20an,and%20 losses%20 surpassing%20%246.9%20 billion.

3. Jibilian, I. (n.d.). *The US is readying sanctions against Russia over the SolarWinds cyber attack. Here's a simple explanation of how the massive hack happened and why it's such a big deal*. Business Insider. Retrieved March 2, 2023, from https://www.businessinsider.com/solarwinds-hack-explained-government-agencies-cyber-security-2020-12

4. Nmsadmin. (2022, September 21). *Why human error is a major threat to cybersecurity in 2022*. NMS Consulting. Retrieved March 2, 2023, from https://nmsconsulting.com/4047/the-human-error-in-cybersecurity/

5. *Cybersecurity framework. NIST. (2023, March 1). Retrieved March 2, 2023, from https://www.nist.gov/cyberframework*

6. *An introduction to the components of the framework*. NIST. (2021, May 14). Retrieved March 2, 2023, from https://www.nist.gov/cyberframework/online-learning/components-framework

7. Force, J. T. (2020, December 10). *Security and Privacy Controls for Information Systems and organizations*. CSRC. Retrieved March 2, 2023, from https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final

8. *Implementing the NIST cybersecurity framework for SMBS*. BIZFORCE Technologies. (n.d.). Retrieved March 2, 2023, from https://bizforcetech.com/implementing-the-nist-cybersecurity-framework-for-smbs/
9. Tara Seals US/North America News Reporter. (2017, January 6). *Organizations struggle with implementing security frameworks*. Infosecurity Magazine. Retrieved March 2, 2023, from https://www.infosecurity-magazine.com/news/organizations-struggle-security/

10. *Cybersecurity framework faqs using the framework*. NIST. (2016, August 25). Retrieved March 2, 2023, from https://www.nist.gov/cyberframework/cybersecurity-framework-faqs-using-framework

11. *A complete guide to NIST cybersecurity framework*. HackControl. (2021, July 31). Retrieved March 2, 2023, from https://hackcontrol.org/blog/nist-cybersecurity-framework/

12. *Automate compliance ISO 27001, SOC 2, PCI DSS*. CyberArrow. (n.d.). Retrieved March 3, 2023, from https://www.cyberarrow.io/

13. Sethi, N. (2022, May 2). *TF-IDF for similarity scores*. Medium. Retrieved March 3, 2023, from https://medium.datadriveninvestor.com/tf-idf-for-similarity-scores-391c3c8788e8

14. Author Valerie Niechai Date Jul 25, & Author Valerie Niechai. (2022, July 25). *TF-IDF tool for SEO - how to guide, Formula & Algorithm*. SEO Software. Retrieved March 3, 2023, from https://www.link-assistant.com/news/tf-idf-tool-for-seo.html

15. Chen, K. (2021, May 24). *Introduction to natural language processing - tf-IDF*. Medium. Retrieved March 3, 2023, from https://kinder-chen.medium.com/introduction-to-natural-language-processing-tf-idf-1507e907c19

16. Stopwords. (n.d.). Retrieved March 3, 2023, from https://www.ranks.nl/stopwords

17. *To-go food packaging and condiments - kari-out*. Kari. (2023, January 23). Retrieved March 3, 2023, from https://kariout.com/

18. Douglas, C. (2021, November 9). *Finding word similarity using tf-idf and cosine in a term-context matrix from scratch in Python*. Medium. Retrieved March 3, 2023, from https://towardsdatascience.com/finding-word-similarity-using-tf-idf-in-a-term-context-matrix-from-scratch-in-python-e423533a407

19. Spencer, T. (2019, November 15). *What is the NIST SP 800-171 and who needs to follow it?* NIST. Retrieved March 3, 2023, from https://www.nist.gov/blogs/manufacturing-innovation-blog/what-nist-sp-800-171-and-who-needs-follow-it-0