Link Prediction for Cybersecurity Companies and Technologies

Santiago Anton Moreno[1]; Dimitri Percia David[2,3,*]; Alain Mermoud[2]; Thomas Maillart[3]; Anita Mezzetti[4]

[1] EPFL, Section of Mathematics
[2] EPFL Cyber-Defence Campus, armasuisse Science and Technology
[3] University of Geneva, Geneva School of Economics and Management, Information Science Institute
[4] EPFL, Section of Financial Engineering
[*] Corresponding Author: dimitri.perciadavid@unige.ch

Abstract

The cybersecurity market is a dynamic environment in which novel entities – technologies and companies – arise and disappear swiftly. In such a fast-paced context, assessing the relations (i.e., links) between those entities is crucial for investment decisions that aims to foster cybersecurity. In this paper, we present a framework for capturing such relations within the Swiss cybersecurity landscape. By using open data, we first model our dataset as a bipartite graph in which nodes are represented by technologies and companies involved in cybersecurity. Then, we use job-openings data to link these two entities. By extracting time series of such graphs, and by using link-prediction methods, we forecast the (dis)appearance of links (and thus relationships) between technologies and companies. We apply several unsupervised learning similarity-based algorithms, a supervised learning method, and finally we select the best method that models such links. Our results show good performance and promising validation of our predicting power. We suggest that our framework is useful for investment decisions in the domain of cybersecurity, as assessing and forecasting links formation and disappearance between companies and technologies enables to shed some light on the rather opaque cybersecurity landscape. Our framework brings decisions-makers a structured tool for more informed investment decisions.

Keywords— technology forecasting; network science; link prediction; time series; supervised learning

1 Introduction

The fast-paced development of technologies reshapes the cybersecurity market [24]. Examples of technologies that redefine cyberdefense are numerous: e.g., quantum computing threatening cryptography protocols, adversarial machine learning, novel communication protocols, behaviour-based authentication of IDS, distributed ledgers. In such a complex technology-development context, both threats and opportunities emerge for actors of the cyberspace [14]. Consequently, a race for a technological advantage takes place between attackers and defenders [20].

The assessment of the cybersecurity technological landscape has become a central activity when it comes to develop cyberdefense strategies [7]. Such an assessment helps defenders to grab an edge in this technological race by developing threat-intelligence tools to reduce the information asymmetry between attackers and defenders [30]. In particular, it enables to foster cyberdefense by identifying more active and developing entities – i.e., technologies and companies – involved in the cybersecurity technological landscape and, then investing in the most relevant ones. Such assessment can be also useful in other ways as it can identify expanding and emerging fields which could gather a lot of investments from public and private sectors since cyberthreats are a great danger for both. Even actors, like venture capitals or angel investors, which seek to invest in risky emerging fields, could benefit greatly from such tools to gain new insights in specific technological fields.

In this work, we aim to contribute to the technological landscape assessment effort by presenting a framework for capturing the relationship between entities of the Swiss cybersecurity technological landscape. By using a dataset coming from the *Technology & Market Monitoring* (TMM) platform, we first model the data as a bipartite graph in which nodes (i.e., entities) are represented by technologies and companies involved in cybersecurity. We then use job openings and keywords from those texts to link entities. By extracting time series of such a graph, and by using link prediction methods, we forecast the (dis)appearance of links between entities. We apply several similarity-based algorithms and a supervised learning machine-learning model that uses outputs from the former. Next we select the best model based on performance measures. We suggest that our framework is useful for any actors who need to forecast cybersecurity technology development or interest in certain companies to assess their security needs, their investments strategy or even their research strategy. Companies could use this method to predict which technology will become the main focus of their competitors and then make sound decisions on their strategy. Adapting this methodology to other technology fields, with other keywords, could be helpful to other quickly evolving markets like fintech and bio-engineering.

The remainder of this paper is structured as follows: Section 2 present the related works; Section 3 presents the theoretical framework and methods employed for link prediction; Section 4 presents the data sources and data structure; Section 5 shows results; Section 6 sets the future work, discuss the limitations and concludes.

2 Related Work

In technology landscape, uncertainty in which technology will emerge and which actors will take actions is always prevalent. Numerous attempts have been devoted to reduce that uncertainty [35]. Early methodology to assess technological opportunity were based on qualitative analysis and on knowledge and experience of fields experts. However, it becomes more and more difficult to assess all the variables in play for experts as the technology environment becomes complex. Early quantitative methodology analyzed technical documents via text mining approaches to provide technological insights [8, 9, 13, 21]. In this work we present a network science quantitative methodology to unravel

technology opportunities and study the actors which enables such technology development. In addition this work will use job openings as its main data which has, to the best of our knowledge, not been analyzed in technology monitoring as a potential indicator. it has been showed that job openings are a great data source to uncover trends in jobs qualification and technological requirements of the employer [17]. Job openings offers a new perspective compared to traditional patent text mining as it highlights a different life period of technology mainly the operational and deployment of a technology. With job openings metrics we can analyze technologies that already have a strong foot in the industry and are already in advanced research or even implemented in companies products, which may reduce the uncertainty of discovering technology opportunities. This data also has the advantage of evolving with creative destruction mechanism since they appear and disappear following offer and demand, unlike patents which never disappear and stay in the data for long periods of time not highlighting the destruction mechanisms. Many complex systems can be studied trough network science frameworks as it summarizes interactions and relationships in a simple and observable way. Link prediction is a method for analyzing links and nodes evolution. It is widely used to predict the appearance and disappearance of links in any kind of networks [33]. In fact, link prediction can be applied as long as the data and the relation between entities is represented via networks. Indeed, studies applying link prediction to various fields emerged such as prediction in healthcare and gene expression networks [2], finding business partners [26] and friends recommendations [1]. By accounting for network structures and other available variables, link prediction methods extract metrics accounting for the likelihood of edges (dis)appearances through time (e.g., [23]).

In this respect, Kim et al. used link prediction to forecast technology convergence [15] using Wikipedia hyperlinks and obtained statistically significant results in the 3D printing example. In addition Lee et al. [21] used F-terms, a patent classification code, to construct a technology network to identify technology opportunities. Kim and Geum [18] developed a data driven technology road map ,using data from patents and specific market trends publications, and created a keyword co-occurrence network on which they used link predictions to detect new opportunities in the road map. In our framework we will obtain bipartite networks since we study two different entities namely technologies and companies. No clear description or performance diagnostic has been given in their work. Regarding bipartite networks, Benchettara et al. [4] adapted link prediction metrics for those networks that enhance performance in many real world data over traditional metrics. Moreover, Silva et al. [27] and Tylenda et al. [32] explored time dependant metrics to use time series within link prediction analysis and showed significant improvement over time ignorant methods. Supervised learning has been applied to link prediction investigations: Mohammad et al. [11] applied supervised learning to a co-authoring network using topological and nodal features for several classification algorithms. More recently, deep learning algorithms were applied to link prediction to further improve performance [13, 37]. Those methods work well when the graphs considered are huge and the performance are in fact similar to classic supervised methods in most cases.

However, to the best of our knowledge, link prediction as a method for assessing the dynamics of the cybersecurity technological landscape has not been explored yet. At least, we found no work focusing on predicting the "birth" and "survivability" of links between entities that compose the cybersecurity technological landscape. Furthermore, job openings are not used as an building block for technological networks and the time evolution of networks is rarely considered. In this work, we present a network-analytics framework that employs link prediction to forecast emerging research and the interest of companies towards cybersecurity technologies. We will use three standard methods of similarity based link prediction and also a supervised learning frameworks which utilizes the aforementioned

methods and output to have a better performance.

3 Theoretical Framework

In this Section, we ground the theoretical knowledge required to apply the methods presented in Section 4.

3.1 Network Science

We define a network $G = (V, E)$, wherein V is any finite set called the vertex set and $E \subseteq V \times V$, called the edge set, corresponds to relation between elements of V. Let $x, y \in V$, such as:

- the neighborhood of x is $\Gamma(x) = \{y \in V s.t.\ (x, y) \in E\}$;
- the degree of x is $\delta_x = |\Gamma(x)|$;
- there is a path between x and y if there exists $(x_0, x_1, ..., x_n)$ such that $x_0 = x\ x_n = y$ and $(x_i, x_{i+1}) \in E\ \forall\ 0 \leq i \leq n-1$;
- a graph G is said to be bipartite if there exists $A, B \subset V$ such that if $(x, y) \in E$ then x and y are not in the same subset A, B.
- Let $|V| = n$, $A \in \mathbf{R}^{n \times n}$ is an adjacency matrix of G if and only if $\forall x, y \in V A_{x,y} = 1$ implies $(x, y) \in E$ and $A_{x,y} = 0$ otherwise.

In this article, V corresponds to companies and technologies and E to links between them. We use the term node or vertex when the distinction between company and technology is not necessary. All the networks considered here are bipartite because a company can only form links with technologies and respectively a technology with companies. We define also $\mathbf{G}$ to be the set containing all the graphs ordered by time meaning $G_0, G_{33} \in \mathbf{G}$ are the graph corresponding to March 2018 and December 2020 respectively. we also define $\mathbf{G_{i-j}}$ with $i < j \in 0, 1, ..., 32$ to be the subset of $\mathbf{G}$ that contains all graph between G_i and G_j (including them). $\mathbf{G_{i-j}}$ will also be called the training graphs in the following sections.

3.2 Link Prediction

In our framework, we define the link prediction problem as follows: Given a set of training graphs $\mathbf{G_{i-j}}$ and a forecast range t (in our case $1 < t < 6$ and it corresponds to months), predict the existence or non existence of a link between every company and every technology in G_{j+t}. Basically we use the graphs in a certain time range to predict the state of the graph t months ahead. Obviously $j + t$ must be smaller than 33. We will also consider the following confusion matrix and notations:

Table 1: **Confusion Matrix**

Actual class \ Predicted class	existing	non-existing
existing	tp	fn
non-existing	fp	tn

Here is the confusion matrix we consider for our problem where existing class refers to actual links in the network and non-existing to links not appearing in the network. tp, fp, fn and tn corresponds respectively to the number of true positives, false positives, false negatives and true negatives. the total number of positives in our problem noted P is equal to $tp + fn$ and thus for the negatives $F = fp + tn$.

The most simple and most studied algorithms in link predictions are based off similarity measures where each pair of nodes, x and y, are given a score s_{xy}, which is defined as the similarity between the nodes. The greater s_{xy} is, the higher the likelihood of link existence is between x and y. In this work, we focus on network topology similarity measures that are based on the network structure only and not on nodes attributes. Given scores s_{xy} for $x, y \in V$ and a threshold θ, we predict the existence of a link between the two nodes if $s_{xy} >= \theta$ and if $s_{xy} < \theta$ the link should not exist. The threshold θ is obtained from training data so that it maximizes a simple function of True Positive Rate (TPR) and False Positive Rate (FPR). In this work we choose to optimize the difference between TPR and FPR which is well known as Youden's J statistic [36]. Other threshold moving techniques are also studied like the geometric mean of sensitivity and specificity and the F score of precision and recall [29] but they nearly always resulted in identical thresholds.

In our framework of time series of graphs, we compute the similarities for each graph in $\mathbf{G}_{i-j}$ and use this data to forecast the similarities for the graph G_{j+t} using several forecasting models. Having the forecast measures and a threshold obtained by the last network in $\mathbf{G}_{i-j}$, we can compute performance diagnostics from G_{j+t}. As the link prediction problem is often a highly unbalanced classification problem we use Receiver Operating Characteristic (ROC) curve as a graphical representation of performance [6, 34]. The ROC curve plots the true positive rate (tpr) against the false positive rate (fpr) and we will use the Area under the ROC curve (AUC) as a metric to optimize. Indeed The AUC is equal to the probability that a classifier will rank a randomly chosen existing link higher than a randomly chosen non existing one so it is less affected by unbalanced classification problems. We also compute Precision Recall curves (PR curves) where $Precision = \frac{tp}{tp+fp}$ and $Recall = tpr$ and compute the Average Precision (AP) which summarizes such a plot as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight. We also note that the AP of a random classifier will should be on average equal to $\frac{P}{P+F}$ and the AUC of such classifier should be around 0.5. The problem with AP as a performance measure is that we can't really compare it using cross validation as its scale varies greatly on the distribution of the test set and thus is sometimes hard to interpret. We applied those diagnostics metrics to the full test set and also to a randomized balanced version of this set and always obtained similar result equal up to the third decimal place.

There is a fair amount of corpus for assessing similarity metrics that are used in link prediction [23], but since we are applying them to bipartite networks, most of the simple local metrics that uses intersection of neighborhood will be of no use [19]. the neighborhood of a company consist of technologies and technology's neighborhood will have companies so no well-know metrics like common neighboors, Adamic-Adar, Jaccard Index will always be equal to 0 despite working well in unipartite networks. For local similarity measure, meaning metrics that uses only information about neighboring vertices, there is only the preferential attachment index which is the simpler and more rapidly computable of all. Then we had to look to global indices that did not require that much computing time. Katz and hyperbolic sine indices were chosen because of the simplification that can be made when working on bipartite graphs and the good approximations we can perform to save running time.

(1) *Preferential Attachment Index* [3]: It is a local index that assumes that the higher the degree of the two nodes considered is, the higher the likelihood of them connecting. It is mathematically defined as:

$$S_{xy}^{PA} = \delta_x \cdot \delta_y \tag{1}$$

Where δ_x and δ_y are the degrees of x respectively y which is defined in Section 3.1.

(2) *Katz Index* [16]: It is a global index that given $x, y \in V$ counts the number of paths (Section 3.1) between x and y and the more paths connects them, higher the probability of them connecting in the future. Mathematically, it is a weighted sums of the number of paths of different length with the weights giving exponentially less impact for longer paths. It can be defined as follows:

$$S_{xy}^{Katz} = \sum_{l=1}^{\infty} \beta^l \cdot |paths_{xy}^{<l>}| = \beta A_{xy} + \beta^2 (A^2)_{xy} + \beta^3 (A^3)_{xy} + \cdots \tag{2}$$

where β is the damping coefficient in $[0, 1]$, $paths_{xy}^{<l>}$ is the set of all path with length l connecting x and y and A is the adjacency matrix of the graph (Section 3.1). It should be noted that β should be smaller than $\frac{1}{\lambda_{max}}$ for the series to converge where λ_{max} is the largest eigenvalue of A.

(3) *Hyperbolic Sine Index* [19]: It is a global index similar to the Katz index in its form but its weights are related to the exponential weights. It can be defined in matrix form as follows:

$$S^{HS} = \sinh(\alpha A) = \sum_{l=0}^{\infty} \frac{\alpha^{1+2i}}{(1+2i)!} A^{1+2i} \tag{3}$$

We also used those metrics as feature for a trained classification algorithm. Each edge is represented as a feature vector consisting of 4 values the three indices presented above and also the number of job openings linking the two entities in this graph. We selected a Support Vector Machine (SVM) [12] as our classification model since we suppose the similarity index would be easily separable in vector space. We implemented several kernel but the linear and the radial basis function (RBF) gave out similar results so we selected the RBF to have more flexibility. We used all the possible edges in the graphs of $\mathbf{G}_{i-j}$ as training sets for the SVM using the computed scores and job openings data.

3.3 Forecast

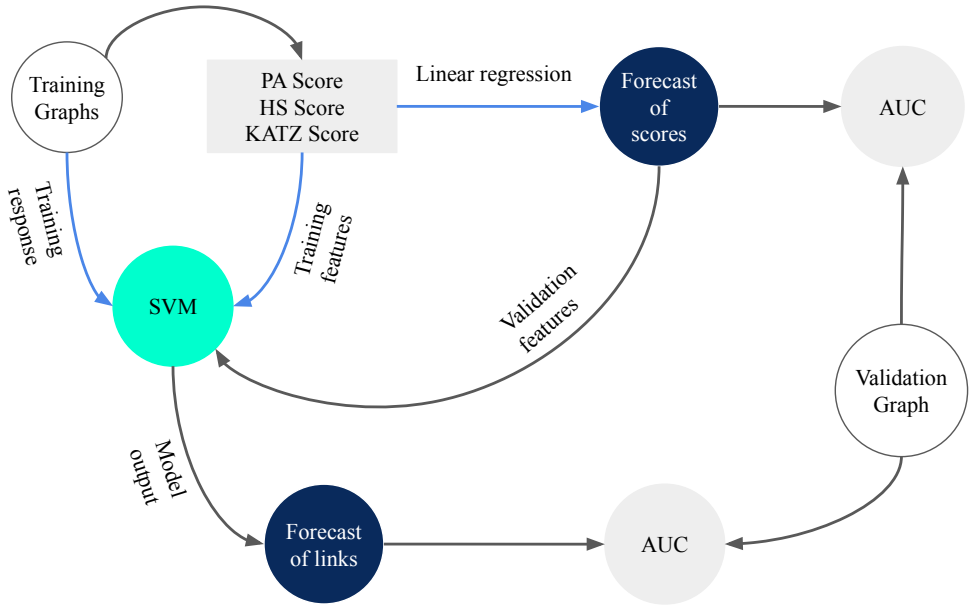
Now assuming we followed the steps described in Section 3.2 with a training set $\mathbf{G_{i-j}}$, we will describe the forecast step of our framework. We have a series of score matrices S_t^{ind} $i < t < j$ and $ind \in \{Katz, PA, HS\}$ for each similarity measure above and a trained SVM model from those scores and other data. Then having a series of scores for each edge S_{xy}^t (we omit the *ind* for clarity but we do this for each similarity measure) we use them to train a time series model to forecast the score at τ steps ahead [25]. We implemented several ARIMA modeling configurations and also a simple linear regression modeling. The ARIMA models took several orders of magnitude more time to compute compared to the linear model and gave out negligible performance enhancement. This seems like a counter intuitive results but Da Silva Soares et al. [27] found similar behaviour for some similarity measures and graphs. Then having trained the linear model, we forecast τ months ahead and use those forecast scores to predict the existence of a link between x and y. We also use those predicted scores as inputs in the trained SVM so that we can test the model.

Now we present the validation methods that we used to obtain robust performance metrics. As presented in Section 3.1 we use ROC/PR curves and AUC/AP as our main performance metrics as the problem is heavily unbalanced. We also computed the accuracy of our methods/models [21, 22, 23, 34]. To compute them in the most robust way we apply blocked cross-validation [5] with block size going from 2 months to 6 months and forecast range from 1 to 6 month. We tried all computations with the untouched unbalanced dataset and with an undersampling method that samples at random the negative class so that we

nearly have a balanced dataset. The ROC curve and AUC seem to not be impacted by the undersampling but the accuracy, PR and AP metrics shifts due to the performance of a random classifier being influenced with class unbalance ratio dropping or increasing. Due to these effects, we do not record the results that used undersampling and only concentrate on the full dataset.

A visual description of all the steps taken in our framework is shown in Figure 1. All this framework was implemented using Python and packages such as Networkx [10], scikit-learn [28] and statsmodel [31].

Figure 1: **Algorithm schema**



Schema of the algorithm steps from training to validation. Blue links corresponds to training steps. there is always at least 2 training graphs and only one validation graph.

4 Data

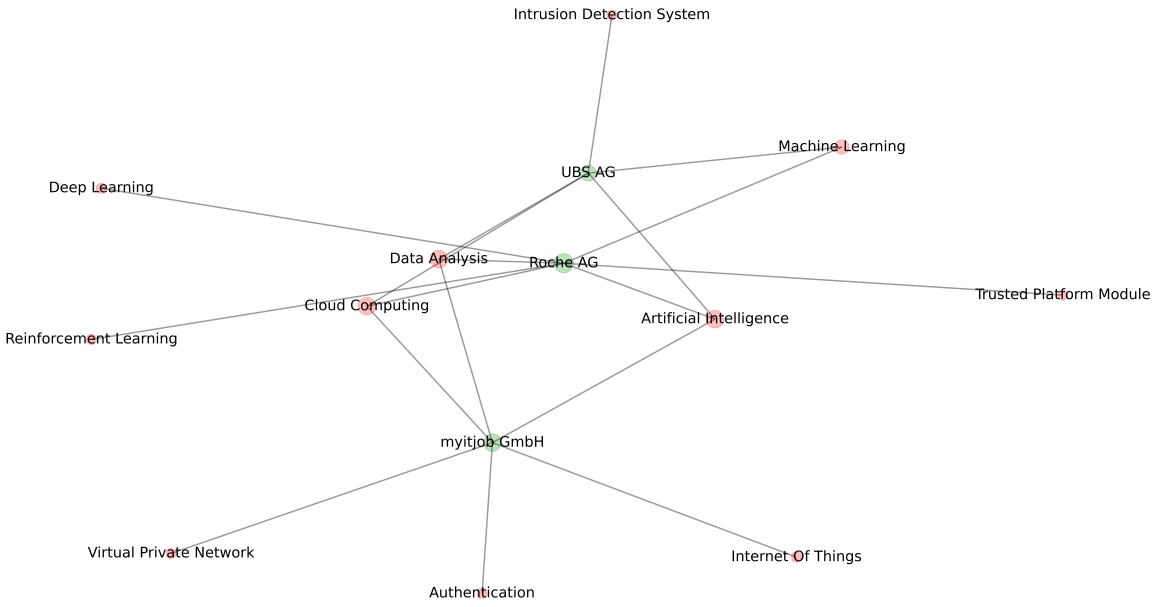
In this section we present the data and the methodology we use to apply the link prediction task.

We use the data collected by the TMM platform (ca. 1 TB) to create a bipartite network composed of technologies and companies of the Swiss cybersecurity technological landscape.[1] The TMM platform is an information system developed by armasuisse Science and Technology (S+T), the Swiss Federal Office for Defence Procurement. TMM aims to exploit big data and open-source information in an automated way for intelligence purposes. The TMM system crawls and aggregates information from different online resources as commercial registers (*Zefix*[2]), websites (*Wikipedia* and *Indeed*[3]) to obtain a list of companies and job openings based in Switzerland.

Job openings are analyzed to obtain keywords of technologies mentioned in their text. By

[1] `https://tmm.dslab.ch`
[2] `https://www.zefix.ch`
[3] `https://indeed.com`

Figure 2: **Subgraph view of the data for November 2019.**



Companies are green and technologies red nodes. The 3 companies depicted are the 3 most linked company in the full graph. Node size is proportional to its degree.

using the companies list and job openings data, we link companies to technology, creating a bipartite network (1805 nodes). You can see part of the network in Figure 2. We also retrieved the number of different job openings that links a company and technology and set it as an edge attribute in the created graph. We use predefined keywords of cybersecurity related technologies to compute word similarity with TMM technologies keywords and select the most relevant. The predefined keywords we used do not represent the full set of all possible cybersecurity technologies as, to the best of our knowledge, there is no consensus on how to obtain such a list in a quantitative way. We then use the difflib[4] library in Python to obtain good matches in the TMM data. We verify the obtained list afterwards to delete any irrelevant matches and thus we obtain 124 keywords[5] from TMM. We talk about the limitations of this keywords selection in Section 6.1. Data , available from March 2018 to December 2020 (34 time-series entries), are crawled from these platforms at different rates, and aggregated monthly.

Cloud computing is the most linked technology throughout the whole time frame. The second is mostly data analysis with a few exceptions in March and April 2018 where machine learning was second. In third place we see three different technologies which are machine learning, Internet of things (IoT), Artificial intelligence (AI). For companies, the most influential in our network are Roche, Novartis and UBS. It is surprising that there is no big tech company in the most connected nodes in the network but analyzing patents data we see that those big tech corporations already have high number of patents citing those technologies. We will further discuss about those findings in Section 6. It is interesting to see that a link when formed has an average of lifespan of 2.8 months before it disappears so

[4] `https://tinyurl.com/8wvkfha2`
[5] keywords link `https://tinyurl.com/jswtsmmn`.

this means that a specific job opening in this field will most likely be filled in this period of time.

5 Results

In this section we will present the performance results of all the methods.

We implemented blocked cross-validation [5] with different training set sizes and forecast range. We can see the mean AUC and accuracy in Table 2. In the forecast range from 1 to 4 month, we observe that longer training set size means higher AUC and accuracy, on the other hand for 5 and 6 month, we observe that the smallest training size gives the best AUC but not the best accuracy. The best accuracy is always obtained by the SVM method with 6 months training size but we should be careful with it as accuracy can be misleading in unbalanced datasets even when undersampling the majority class.

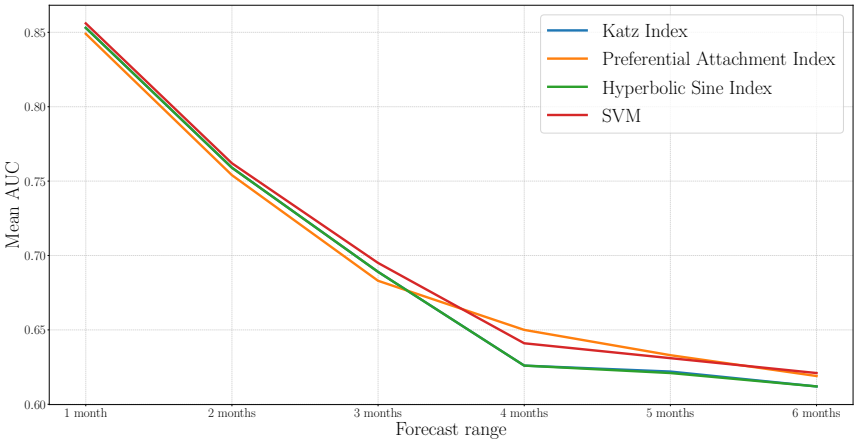
Table 2: **AUC and Accuracy**

			Forecast range					
Method	Training size	Metric	1 month	2 month	3 month	4 month	5 month	6 month
Katz Index	2 months	AUC	0.831	0.737	0.668	0.642	0.628	0.626
		Accuracy	0.852	0.768	0.707	0.683	0.671	0.665
	3 months	AUC	0.836	0.734	0.664	0.632	0.626	0.619
		Accuracy	0.859	0.772	0.711	0.685	0.676	0.669
	4 months	AUC	0.844	0.727	0.652	0.623	0.615	0.606
		Accuracy	0.864	0.772	0.71	0.686	0.676	0.667
	6 months	AUC	0.853	0.759	0.689	0.626	0.622	0.612
		Accuracy	0.871	0.792	0.735	0.693	0.684	0.676
PA Index	2 months	AUC	0.818	0.709	0.643	0.61	0.586	0.574
		Accuracy	0.826	0.743	0.687	0.659	0.646	0.634
	3 months	AUC	0.837	0.733	0.652	0.616	0.599	0.589
		Accuracy	0.837	0.757	0.692	0.668	0.657	0.647
	4 months	AUC	0.844	0.736	0.658	0.62	0.602	0.594
		Accuracy	0.842	0.761	0.703	0.675	0.661	0.653
	6 months	AUC	0.849	0.754	0.683	**0.65**	0.633	0.619
		Accuracy	0.841	0.77	0.718	0.694	0.681	0.672
HS Index	2 months	AUC	0.832	0.737	0.669	0.642	0.628	0.626
		Accuracy	0.853	0.768	0.707	0.683	0.671	0.665
	3 months	AUC	0.837	0.734	0.664	0.632	0.625	0.619
		Accuracy	0.859	0.772	0.711	0.685	0.676	0.669
	4 months	AUC	0.844	0.727	0.652	0.624	0.615	0.605
		Accuracy	0.865	0.772	0.71	0.686	0.676	0.667
	6 months	AUC	0.853	0.759	0.689	0.626	0.621	0.612
		Accuracy	0.871	0.792	0.735	0.693	0.684	0.676
SVM	2 months	AUC	0.836	0.742	0.676	0.648	**0.635**	**0.629**
		Accuracy	0.856	0.773	0.712	0.688	0.678	0.671
	3 months	AUC	0.838	0.741	0.667	0.638	0.63	0.624
		Accuracy	0.863	0.777	0.715	0.69	0.68	0.673
	4 months	AUC	0.852	0.734	0.658	0.628	0.619	0.613
		Accuracy	0.867	0.777	0.715	0.69	0.682	0.673
	6 months	AUC	**0.856**	**0.762**	**0.695**	0.641	0.631	0.621
		Accuracy	0.874	0.794	0.74	0.702	0.692	0.684

Mean AUC and Accuracy obtained through blocked cross-validation for the with given training size and forecasting range. The old used to compute accuracy are obtained by optimizing Younden's J statistic. the standard deviation for those means are between 0.03 and 0.07. The best AUC for a given forecast range is highlighted in bold. The best accuracy is highlighted in red.

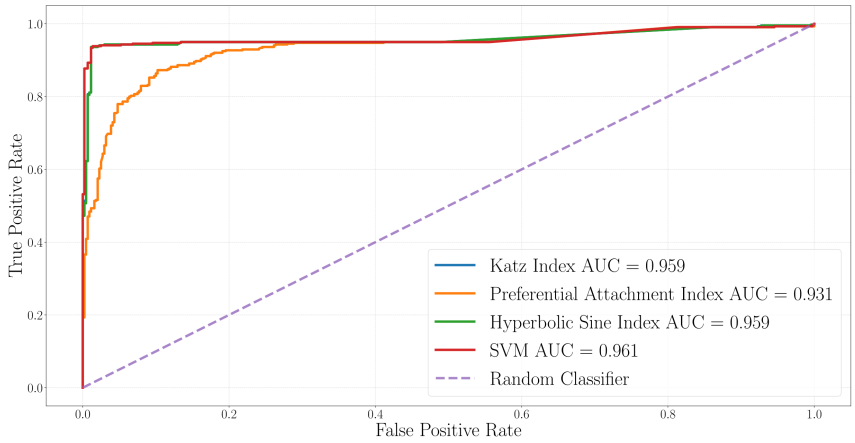
Overall Table 2 shows that our methods are effective and reliable to identify the evolution of the network even for 6 month ahead predictions. In particular, SVM is found to increase the performance of the three indices in most scenarios but there is few exceptions. In fact PA index the simplest of all the metrics has the best AUC for 4 month ahead prediction. A clearer representation of the drop in performance with respect to the forecast range can be seen in Figure 3. The rate of the drop decreases over time which suggests that those methods can still be better than random guess for longer forecast range. Using all the data available we can obtain stronger results. In Figure 4, we used all the data in $\mathbf{G_{0-32}}$ to predict the links in G_{33}. This ROC curve shows great performance and we also get an accuracy of 0.965 for this setting. Forecasting G_{33} with $\mathbf{G_{0-28}}$ results in AUC between 0.743 and 0.753 for all methods and an accuracy between 0.75 and 0.77 which is much greater than the mean AUC and accuracy for all training size and that forecast range. This illustrates the high volatility of those performance results which highly depends on the test set.

Figure 3: **Mean AUC evolution with forecast range**



Mean AUC of all the methods against forecast range. These means have been calculated through blocked cross-validation using training sets of size 6. Standard deviation is between 0.03 and 0.07.

Figure 4: **ROC Curve**

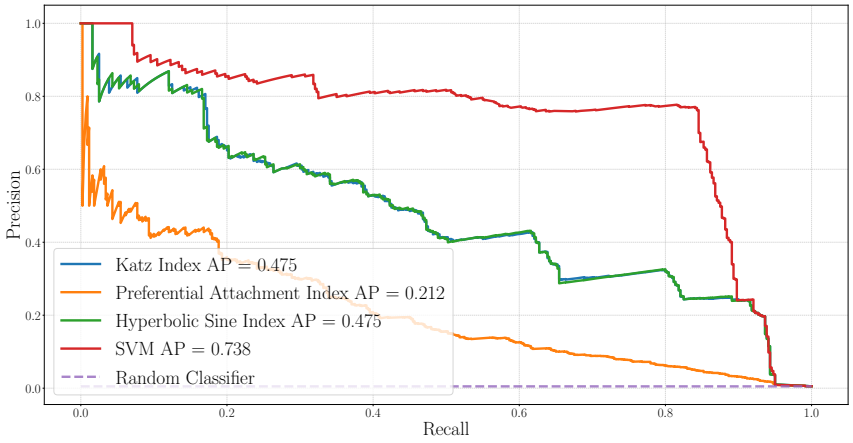


ROC curve of the different methods with $\mathbf{G_{0-32}}$ as training set and tested on G_{33}.

Now looking at the Precision Recall curves and AP values in Figure 5, which uses the same

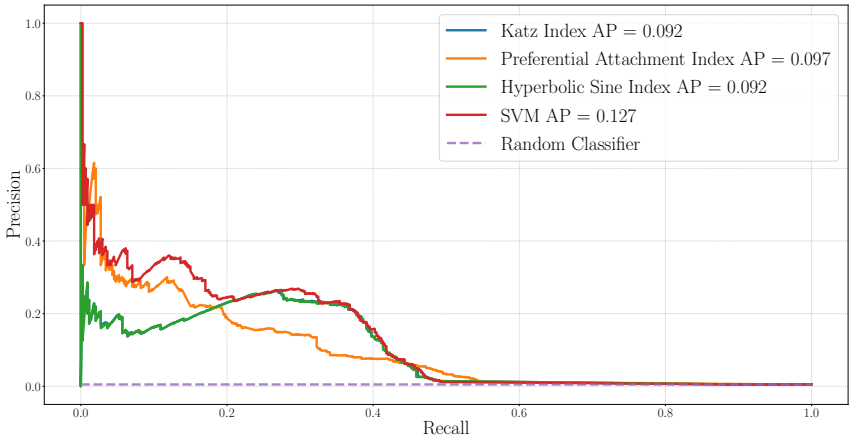
training and testing set as Figure 4, We see that overall the same ranking of methods arises. On the other hand, the difference in AP value between the methods is a lot more obvious compared to the AUC. Even with other training size and testing sets, The SVM is clearly the best performer. The Katz and Hyperbolic Sine index are more or less similar, even though in some curves we see more differentiation between the two that did not appear in the ROC curves. And finally, The PA index is the least performer in the majority of tests by the AP metric. Predicting further in time and using a smaller training set decreases the AP values and flattens the Precision Recall curves drastically but it is still quite above the random classifier performance as we can see in Figure 6.

Figure 5: **Precision Recall Curve**



Precision Recall curve of the different methods with $\mathbf{G_{0-32}}$ as training set and tested on G_{33}. The AP of a random classifier is less than 0.01.

Figure 6: **Precision Recall Curve**



Precision Recall curve of the different methods with $\mathbf{G_{21-27}}$ as training set and tested on G_{33}. The AP of a random classifier is less than 0.01.

6 Conclusion

In this work we presented a network science framework for technology monitoring and forecast. It proposes a systematic approach to anticipate companies interest in cybersecurity technologies and help investors or cybersecurity actors make decisions on quantitative insights. For this purpose a bipartite companies-technologies network was created using open data from market registry platforms and job openings from *Indeed* to form links between entities. Next, four link predictions methods were used to predict which technologies will be of interest for companies. We implemented three classic similarity based algorithms and also an supervised SVM method to our data. This study of this environment on Swiss companies confirms that the proposed method enables to forecast emerging interest of companies in technologies with statistically significant results. We hope this work strengthen the network science approach for technology monitoring and also entice researcher to deepen the study of job openings text mining as an indicator of development and an alternative to standard patent and technical text analysis. We also suggest that this frameworks can be extended to any other technological fields and even become adaptable according to the customer requirements.

6.1 Limitations

To better see the feasibility and utility of our framework we need to collect more data from other sources in other countries as the Swiss cybersecurity landscape is but a small fraction of the whole environment. We would need to set up a bigger crawling protocol and adapt the keywords extraction done by TMM to other languages. We also need a clear disambiguation framework for companies as many have multiple market registry entries for in fact the same entity. We also need to obtain, with a quantitative methodology, consistent technology keywords with specialized category matching so that the networks that we create can be custom made for specific industries and fields.
Another question is the validity of job openings as an indicator for technology opportunity. We could also broaden the search for job openings as many such proposals are also posted on other platforms such as LinkedIn, Twitter and other social media.

6.2 Further Research

Future works should focus on broadening the data and adapting the data crawling steps to other platforms, languages and keywords. Furthermore, We could also customize the methodology to several technology fields so that investors or other actors could obtain insights in their desired environment. As the data becomes bigger, one should also investigate other link predictions methods like neural networks and maximum likelihood probabilistic models as they seem to obtain stronger results the bigger the networks become compared to similarity based methods. Graphs embedding should also be considered as it is a very active field and can represents graphs and nodes in a suitable format for many classification models. Working with supervised learning methods, Researcher might also look to extract other features to train such models like job openings sentiment analysis metrics, count of keywords in text as well as indicators from other sources of data (scientific publications, tweets, patents, market data, etc...). If this data is available, a specialized similarity metric could also be developed using those indicators to set up an unsupervised model since this has been showed to increase performance on lots of social networks. Since our project is not subject to the laws of the market it's real value and efficacy should be judged by the companies and investors who could do portfolio analysis using results obtained from this methodology.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.
Declarations of interest: none

References

1. Akcora, C. G., Carminati, B. & Ferrari, E. *Network and profile based measures for user similarities on social networks* in *2011 IEEE International Conference on Information Reuse Integration* (Aug. 2011), 292–298. doi:10.1109/IRI.2011.6009562.
2. Almansoori, W. *et al.* Link prediction and classification in social networks and its application in healthcare and systems biology. en. *Network Modeling Analysis in Health Informatics and Bioinformatics* **1,** 27–36 (June 2012).
3. Barabási, A.-L. & Albert, R. Emergence of Scaling in Random Networks. en. *Science* **286.** Publisher: American Association for the Advancement of Science Section: Report (1999).
4. Benchettara, N., Kanawati, R. & Rouveirol, C. *Supervised Machine Learning Applied to Link Prediction in Bipartite Social Networks* in *2010 International Conference on Advances in Social Networks Analysis and Mining* (2010).
5. Bergmeir, C. & Benítez, J. M. On the use of cross-validation for time series predictor evaluation. en. *Information Sciences. Data Mining for Software Trustworthiness* **191,** 192–213 (May 2012).
6. Fawcett, T. An introduction to ROC analysis. en. *Pattern Recognition Letters. ROC Analysis in Pattern Recognition* **27,** 861–874 (June 2006).
7. Fleming, T. C., Qualkenbush, E. L. & Chapa, A. M. The Secret War Against the United States:The Top Threat to National Security and the American Dream Cyber and Asymmetrical Hybrid Warfare An Urgent Call to Action. *The Cyber Defense Review* **2.** (2021) (2017).
8. Gerken, J. M. & Moehrle, M. G. A new instrument for technology monitoring: novelty in patents measured by semantic patent analysis. en. *Scientometrics* **91,** 645–670 (June 2012).
9. Geum, Y., Kim, C., Lee, S. & Kim, M.-S. Technological Convergence of IT and BT: Evidence from Patent Analysis. en. *ETRI Journal* **34,** 439–449 (2012).
10. Hagberg, A., Swart, P. & S Chult, D. *Exploring network structure, dynamics, and function using networkx* English. Tech. rep. LA-UR-08-05495; LA-UR-08-5495 (Jan. 2008).
11. Hasan, M. A., Chaoji, V., Salem, S. & Zaki, M. Link Prediction using Supervised Learning. en (Jan. 2006).
12. Hearst, M., Dumais, S., Osuna, E., Platt, J. & Scholkopf, B. Support vector machines. *IEEE Intelligent Systems and their Applications* **13.** Conference Name: IEEE Intelligent Systems and their Applications, 18–28 (July 1998).
13. Huang, L. *et al.* Tracking the dynamics of co-word networks for emerging topic identification. en. *Technological Forecasting and Social Change* **170,** 120944. doi:10.1016/j.techfore.2021.120944 (Sept. 2021).
14. Jang-Jaccard, J. & Nepal, S. A survey of emerging threats in cybersecurity. en. *Journal of Computer and System Sciences* **80** (2014).
15. Juram Kim, Seungho Kim & Lee, C. Anticipating technological convergence: Link prediction using Wikipedia hyperlinks. en. *Technovation* **79** (2019).

16. Katz, L. A new status index derived from sociometric analysis. en. *Psychometrika* **18,** 39–43. doi:10.1007/BF02289026 (Mar. 1953).

17. Kim, J. & Angnakoon, P. Research using job advertisements: A methodological assessment. en. *Library & Information Science Research* **38,** 327–335. doi:10.1016/j.lisr.2016.11.006 (Oct. 2016).

18. Kim, J. & Geum, Y. How to develop data-driven technology roadmaps:The integration of topic modeling and link prediction. en. *Technological Forecasting and Social Change* **171,** 120972. doi:10.1016/j.techfore.2021.120972 (Oct. 2021).

19. Kunegis, J., De Luca, E. W. & Albayrak, S. *The Link Prediction Problem in Bipartite Networks* en. in *Computational Intelligence for Knowledge-Based Systems Design* (eds Hüllermeier, E., Kruse, R. & Hoffmann, F.) (Springer, 2010). doi:10.1007/978-3-642-14049-5_39.

20. Laube, S. & Böhme, R. Strategic Aspects of Cyber Risk Information Sharing. en. *ACM Computing Surveys* **50.** doi:10.1145/3124398 (2017).

21. Lee, J., Ko, N., Yoon, J. & Son, C. An approach for discovering firm-specific technology opportunities: Application of link prediction to F-term networks. en. *Technological Forecasting and Social Change* **168,** 120746. doi:10.1016/j.techfore.2021.120746 (July 2021).

22. Lichtenwalter, R. N., Lussier, J. T. & Chawla, N. V. *New perspectives and methods in link prediction* in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (Association for Computing Machinery, July 2010), 243–252. doi:10.1145/1835804.1835837.

23. Linyuan Lü & Tao Zhou. Link prediction in complex networks: A survey. en. *Physica A: Statistical Mechanics and its Applications* **390,** 1150–1170. doi:10.1016/J.PHYSA.2010.11.027 (Mar. 2011).

24. Lundstrom, M. APPLIED PHYSICS: Enhanced: Moore's Law Forever? *Science* **299,** 210–211 (2003).

25. Montgomery, D. C. Introduction to Time Series Analysis and Forecasting. en. *Wiley Series in Probability and Statistics* **526,** 5. doi:10.1126/SCIENCE.1079567 (Jan. 2008).

26. Mori, J., Kajikawa, Y., Kashima, H. & Sakata, I. Machine learning approach for finding business partners and building reciprocal relationships. en. *Expert Systems with Applications* **39,** 10402–10407. doi:10.1016/J.ESWA.2012.01.202 (Sept. 2012).

27. P. R. da Silva Soares & R. B. C. Prudêncio. *Time Series Based Link Prediction* in *The 2012 International Joint Conference on Neural Networks (IJCNN)* (2012). doi:10.1109/IJCNN.2012.6252471.

28. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. en. *MACHINE LEARNING IN PYTHON,* 6 (Oct. 2011).

29. Powers, D. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. *Mach. Learn. Technol.* **2** (Jan. 2008).

30. Qamar, S., Anwar, Z., Rahman, M. A., Al-Shaer, E. & Chu, B.-T. Data-driven analytics for cyber-threat intelligence and information sharing. en. *Computers & Security* **67.** doi:10.1016/J.COSE.2017.02.005 (2017).

31. Seabold, S. & Perktold, J. *Statsmodels: Econometric and Statistical Modeling with Python* en. in (2010), 92–96. doi:10.25080/MAJORA-92BF1922-011.

32. Tylenda, T., Angelova, R. & Bedathur, S. *Towards time-aware link prediction in evolving social networks* in (Association for Computing Machinery, 2009). doi:10.1145/1731011.1731020.

33. Wang, P., Xu, B., Wu, Y. & Zhou, X. Link prediction in social networks: the state-of-the-art. en. *Science China Information Sciences* **58,** 1–38. doi:10.1007/S11432-014-5237-Y (Jan. 2015).

34. Yang, Y., Lichtenwalter, R. N. & Chawla, N. V. Evaluating link prediction methods. en. *Knowledge and Information Systems* **45.** doi:10.1007/S10115-014-0789-0 (2015).

35. Yoon, J. *et al.* Technology opportunity discovery (TOD) from existing technologies and products: A function-based TOD framework. en. *Technological Forecasting and Social Change* **100,** 153–167. doi:10.1016/J.TECHFORE.2015.04.012 (Nov. 2015).

36. Youden, W. J. Index for rating diagnostic tests. en. *Cancer* **3,** 32–35. doi:10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3 (1950).

37. Zhang, M. & Chen, Y. Link Prediction Based on Graph Neural Networks. *arXiv:1802.09691 [cs, stat].* arXiv: 1802.09691. (2021) (Nov. 2018).