

# Knowledge Base pattern structures-based Classification of underground forums: A case study

Abdulrahim Ghazal<sup>1</sup>

<sup>1</sup>National Research University Higher School of Economics, Pokrovsky boulevard, 11, 109028, Moscow, Russian Federation

## Abstract

Underground forums messages are online platforms where hackers share information and tools for cyber-attacks. This paper discusses using a knowledge base in the context of lazy classification of underground forums messages using pattern structures to assess the risk of these messages. Comparing the performance of pattern structures and the knowledge base approach shows a significant improvement in time needed for classification without loss in accuracy.

## Keywords

Formal concept analysis (FCA), Threat intelligence, Underground forums, Pattern structures, Knowledge Bases

## 1. Introduction

Corporations and organizations have been under increasing level of cyber attacks happening in variety of intensity, frequency and impact [1]. This increase has made collecting information and analyzing findings about these attacks more vital. Threat Intelligence is the practice that focuses on that, and provides insights to victims on the history, current state of the attacks and what to do to mitigate them [2]. The field of Threat Intelligence has been growing a lot in the past years due to the business and regulations needs for more aware cybersecurity practices.

The adequate implementation of such service includes monitoring, detecting, analyzing and reporting cyber threats in a timely manner. Sources of information include public and private underground communities (forums) where the attackers share information about their tools, findings and results.

Underground hackers forums are social platforms that host a group of topics with comments from members of the forum. They consist of sub-forums each focusing on a specific sub-field related to cyber crime. While some of these forums include sections for sales of illegal physical goods, threat intelligence focuses on cyber threats only.

Some underground forums are public (like the forum in Figure 1), but many require registration and in some cases references or payment that ranges from 50\$-1000\$. These payments are used sometimes to upgrade a user's status in the forum (VIP, Golden, etc.), and the user would

---


Published in Sergei O. Kuznetsov, Amedeo Napoli, Sebastian Rudolph (Eds.): *The 11th International Workshop "What can FCA do for Artificial Intelligence?", FCA4AI 2023, co-located with IJCAI 2023, August 20 2023, Macao, S.A.R. China, Proceedings*, pp. 07–15.

✉ agazal@hse.ru (A. Ghazal)

ORCID 0000-0003-3873-8513 (A. Ghazal)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

be able to access all sections of the forum. These payments are usually done via cryptocurrency. These forums support "escrow" services with guarantees for deals that are done on the forum.

Collecting the relevant information is the first step to perform impactful threat intelligence, which is followed by detecting threats inside the collected information, while the threat is fresh, or even better, before it occurs. Such process is usually preformed by human analysts who have to go through a large amount of posted messages daily. This effort grows every day with more forums and more messages.

This work aims to assist the analysts in their task and automate the process of detecting information about threats, giving them more time to move on to the next phases of analysis and reporting threats to the relevant authorities or victims.

This will be done by using natural language analysis of the messages with formal concept analysis and its extension pattern structures to classify messages into risky or none risky. The method used should be fast enough to catch up with the incoming stream of messages, and carry some ability to provide a simple explanation of the result of classification.

In this paper, we focus on using knowledge bases built from previous training iterations of lazy classification using pattern structures, and comparing the knowledge base with the raw application of pattern structures lazy classification.

The rest of the paper is organized as follows: In Section 2 we recall basic definitions in formal concept analysis, pattern structures and lazy classification method using pattern structures. In Section 3 we describe the experimental setting and knowledge base building. In Section 4, we discuss the preliminary results of applying the knowledge base approach in the context of lazy classification to underground forum messages. We conclude the work in section 5.

## 2. Formal Concept Analysis

### 2.1. Main Definitions

Formal Concept Analysis (FCA) is a mathematical theory that is based on concepts and conceptual hierarchy [3, 4]. Its structuring of knowledge representation was used for knowledge discovery, data analysis [5, 6], mining association rules [5, 7] ontology design [8, 9], and recommendation systems [10].

### 2.2. Pattern Structures

This extension of Formal Concept Analysis was developed as an effort to enable applying the mathematical tools offered by Formal Concept Analysis with a more complex data structures like graphs, non-binary or vector data [11].

Let  $G$  be a set of objects and  $(D, \sqcap)$  be a meet-semi-lattice of possible object descriptions or patterns (for standard FCA, it would be the powerset of attribute set) with the similarity operator  $\sqcap$ . Elements of  $D$  are ordered by a subsumption relation  $\sqsubseteq$  such that  $a, b \in D$ , then one has  $a \sqsubseteq b \Leftrightarrow a \sqcap b = a$ . We also define  $\delta : G \rightarrow D$  as a mapping between objects and their attributes. We call  $(G, \underline{D}, \delta)$  where  $\underline{D} = (D, \sqcap)$  a pattern structure. We can define the operators  $(\cdot)^\circ$  on  $A \subseteq G$  and  $d \in (D, \sqcap)$  making Galois connection between the powerset of objects and

---

**Algorithm 1: Lazy Classification with Pattern Structures**

---

Requires: pattern structure  $(G, \underline{D}, \delta)$ , test example  $g_t \in G_\tau$  with description  $\delta(g_t)$ , parameter  $0 \leq \alpha \leq 1$ .

1: for  $g \in G_+ \cup G_-$  :

2: compute  $\text{sim} = \delta(g) \sqcap \delta(g_t)$

3:  $\text{extsim} = (\text{sim})^\diamond$

4: if  $\alpha\%$  of objects in  $\text{extsim}$  have target attribute, classify  $g$  positive

5: if  $\alpha\%$  of objects in  $\text{extsim}$  do not have target attribute, classify  $g$  negative

6: classify undetermined (the algorithm terminates without classification).

---

ordered set of descriptions:

$$A^\diamond = \sqcap_{g \in A} \delta(g) \quad (1)$$

$$d^\diamond = \{g \in G \mid d \sqsubseteq \delta(g)\} \quad (2)$$

These operators will give us back the maximal set of patterns shared by the objects in  $A$  and the maximal set of objects that share the description  $d$ , respectively.

A pair  $(A, d)$ ,  $A \in G$  and  $d \in (D, \sqcap)$  that satisfies  $A^\diamond = d$  and  $d^\diamond = A$  is called a *pattern concept*, where  $A$  is called the *extent* and  $d$  is called the *pattern intent* of  $(A, d)$ .

A partial order  $\leq$  is defined on the set of concepts:  $(A, d_1) \leq (B, d_2)$  iff  $A \subseteq B$  (or, equivalently,  $d_2 \sqsubseteq d_1$ ). This partial order forms a complete lattice on the set of all pattern concepts. We call this the pattern concept lattice of the pattern structure  $(G, \underline{D}, \delta)$ .

For classification tasks we do not need to extract the full hidden knowledge from a dataset in terms of implications, hypotheses or association rules, but a so-called lazy classification can be applied [12, 13].

### 2.3. Lazy Classification with Pattern Structures

In classification problems we have a target attribute, which, in the simplest case of two classes, has two values, denoted by  $+$  and  $-$ . By  $G_+$  we denote the set of objects that have the target attribute (positive examples) and by  $G_-$  we denote the set of objects that do not have the target attribute (negative examples), so that  $G_+ \cap G_- = \emptyset$ . Elements of  $G$  that do not belong to any of these subsets are called unclassified examples  $G_\tau$ .

A version of the lazy classification method [12, 13] is described in Algorithm 1.

This algorithm takes  $O(|G| (p(\sqcap) + |G| p(\sqsubseteq)))$  time, where  $p(\sqcap)$ ,  $p(\sqsubseteq)$  are times for computing  $\sqcap$ ,  $\sqsubseteq$ , respectively.

### 2.4. Knowledge Bases

With the increase of data sizes used to perform some information retrieval or computation on the stored data, it becomes challenging to generate results in a timely manner, which led to the creation of knowledge bases that store some previously proven information that can help in these tasks for fast access [14].

In the context of FCA and pattern structures, concept lattices can offer a new method of knowledge representation [15] and in this work, it will be applied to save time by testing new objects versus well-performing classifiers that are saved from past testing iterations first.

### 3. Experiments

The goal of experiments is to test how using a knowledge base will change the results of the raw application of the pattern structures lazy classification scheme (both in time and performance). To do that, we will need to test several settings of the pattern structures lazy classification scheme with different parameters, then repeat the best performing experiments, but with using the knowledge base instead.

We will perform several experiments starting with lazy classification using the traditional FCA approach, then use the interval, min and max pattern structures. We will repeat the same experiments, but with probabilistic relaxation. In the end, we will repeat the experiments with best performing parameters, but with the use of the knowledge base, to measure the improvement in time needed for classification.

#### 3.1. Dataset

The used dataset is composed of text messages posted by hackers on several underground forums starting from 2021 Provided by the cybersecurity firm F.A.A.C.T. The positive examples of the dataset are real threats detected by human analysts and reported upon. The dataset is balanced in terms of classes and has 4945 messages. These messages come from a core of real threat messages that were selected by the Threat Intelligence analysts team in the aforementioned company. The negative sample was obtained from the same set of underground forums which contained the positive samples, and posted in the same time frame.

The dataset is then processed into a numerical dataset for the later stages, with the use of tf-idf. We should note that the number of keywords that will be included in the results of tf-idf is a parameter that will be controlled during the experiments and is called “min\_df”. It represents the threshold of percentage of documents at which a keyword is included in the vocabulary. We will also control the tolerance factor  $\alpha$  which represents the probabilistic relaxation allowed for counter examples (See Algorithm 1).

#### 3.2. Assessment

we should note that the used version of the lazy classification algorithm allows for unclassified cases, and in this spirit, we define “**Saved Effort**” measure, which is computed as  $1 - \frac{|G_{uncl}|}{|G_\tau|}$ , where  $G_{uncl}$  is the set of unclassified examples  $G_{uncl} \subseteq G_\tau$ .

#### 3.3. Experiments

In all the following experiments, 5-cross validation was used. The code used to perform these experiments is written in python 3.6 and no Off-The-Shelf tools were used to write the FCA or Pattern Structures code, but sk-learn package was used to build the Machine Learning models in

the later sections. Due to space limitations, we present the numerical results (publicly accessible) at: <https://github.com/abdulrahimGhazal/FCA-LC-KB>.

### 3.3.1. Binary Attributes

The attribute values here are the tf-idf values for the keywords contained in the text that were included in the vectorizer's vocabulary resulting from tf-idf. It would be represented as:

$$att\_value(keyword) = \begin{cases} 1 & keyword \in vectorizer\ vocab \\ 0 & otherwise \end{cases} \quad (3)$$

We tested 5 values of min\_df and the highest F1 value was 0.98 and saved effort at 0.88 with min\_df at 0.01. We repeat the same experiment, but with introducing the tolerance factor, allowing for a small amount of counter examples. We get the highest F1 value 0.98 and saved effort 0.92 with min\_df at 0.01, and  $\alpha$  at 75%.

### 3.3.2. Interval Pattern Structure

We represent the values of tf-idf as intervals of the floating point value, such that if the tf-idf value for a keyword is  $x$ , then the attribute value would be an interval  $[x, x]$ . the intersection operator for this pattern structure is defined as:

$$[a_1, b_1] \cap [a_2, b_2] = [min(a_1, a_2), max(b_1, b_2)] \quad (5)$$

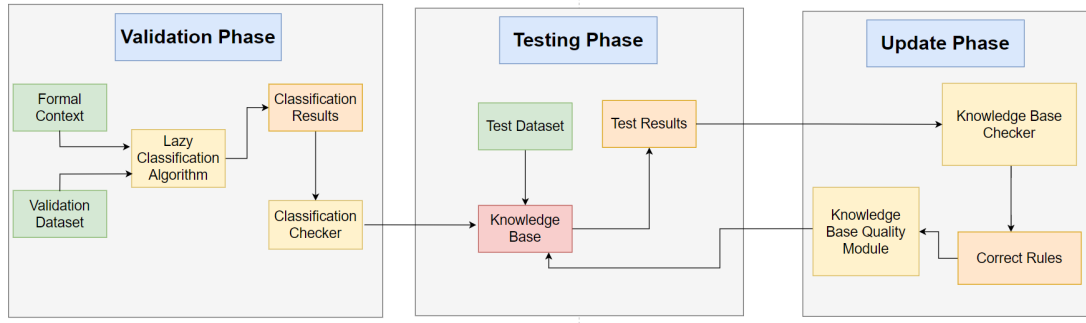
We tested 5 values of min\_df and the highest F1 value was 0.88 and saved effort at 0.88 with min\_df at 0.01. We repeat the same experiment, but with introducing the tolerance factor, allowing for a small amount of counter examples. We get the highest F1 value 0.94 and saved effort 0.94 with min\_df at 0.01, and  $\alpha$  at 75%.

### 3.3.3. Min Pattern Structure

We represent the values of tf-idf as intervals of the floating point value, such that if the tf-idf value for a keyword is  $x$ , then the attribute value would be an interval  $[x, \infty]$ . the intersection operator for this pattern structure is defined as:

$$[a_1, \infty] \cap [a_2, \infty] = [min(a_1, a_2), \infty] \quad (6)$$

We tested 5 values of min\_df and the highest F1 value was 0.90 and saved effort at 0.87 with min\_df at 0.01. We repeat the same experiment, but with introducing the tolerance factor, allowing for a small amount of counter examples. We get the highest F1 value 0.94 and saved effort 0.94 with min\_df at 0.01, and  $\alpha$  at 75%.



**Figure 1:** Knowledge Base building and updating.

### 3.3.4. Max Pattern Structure

We represent the values of tf-idf as intervals of the floating point value, such that if the tf-idf value for a keyword is  $x$ , then the attribute value would be an interval  $]-\infty, x[$ . the intersection operator for this pattern structure is defined as:

$$]-\infty, a_1] \cap ]-\infty, a_2] = ]-\infty, \max(a_1, a_2)] \quad (7)$$

We tested 5 values of min\_df and the highest F1 value was 0.88 and saved effort at 0.88 with min\_df at 0.01. We repeat the same experiment, but with introducing the tolerance factor, allowing for a small amount of counter examples. We get the highest F1 value 0.94 and saved effort 0.94 with min\_df at 0.01, and  $\alpha$  at 75% and 80%.

### 3.3.5. knowledge base experiments

After noticing that a lot of examples are classified using a limited set of attribute sets (from the intersection of new example and randomly selected examples from the objects set), so a knowledge base from all “classifiers” which are basically sets of attributes, that performed well was created. A diagram of the process of building and maintaining the knowledge base can be seen in Figure 1.

This knowledge base was then checked manually by human analysts (experts) to review whether the attributes really carried some truth in the classification. We set the threshold for adding the classifier to the knowledge base as that it must classify correctly 90% of the test examples assigned to it, and it must classify at least 2% of the test examples.

The building of the knowledge base is done via using a validation dataset that is part of the data, with a 5 cross validation process. the classification results then are given to the “classification checker” module, which checks if a classifier was able to classify correctly. If that was the case, the classifier would be added to the interim knowledge base which would be checked in the end of the validation process to trim the classifiers that do not match our predefined conditions.

In case we do not want to update the knowledge base, The resulting classifiers (ready knowledge base) will be passed on to the test dataset, which the classifiers never saw before to

avoid over-fitting. If the results are satisfactory, we stop the process and save the knowledge base to use in real-world cases.

If an update is needed, we also created a monitoring module to update the knowledge base, after each iteration of testing, so that these conditions are not broken by old classifiers, and in the same time, we should maintain the time advantage that the knowledge base with a specific set of classifiers have. The process of updates pass the ready knowledge base to the "knowledge base checker" module which checks if there are any changes in the rules from the old version of the knowledge base, and keeps lists of the removed rules and the added rules.

These rules are then passed to the "knowledge base quality module" which works by making several intermediate steps towards minimizing the changes while keeping the thresholds of quality. These intermediate steps would mean breaking some of the quality thresholds, but only for the removed rules, and only in case the threshold of the number of classified examples (2%) is not achieved. This is due to the new test dataset. The resulting rules then make the final updated knowledge base.

**Binary Rules** In the following experiments, we only depend on the existence of the set of attributes as a classifier, so when using the knowledge base, we only check if the message contains the set of attributes in its text.

Looking at the results, while the best time was observed in traditional FCA before using the knowledge base and after using it at 0.03 milliseconds on average, we notice that the time needed for the rest of tested pattern structures has decreased significantly from more than one second to 0.04 milliseconds, with a slight loss in performance.

**Conditioned Rules** Unlike the previous experiment, we save the values of the attributes in the set of the attributes of the to-be-added classifiers, then we apply the intersection of the corresponding pattern structure on the whole test cases.

Since the binary attributes already mean that the keyword exists or not, the results will not change for the binary attributes experiments. For the pattern structure experiments, we notice that the loss of accuracy has decreased, but the time needed has increased slightly, since we have to check more rules before reaching a classification, but still much better than going through the usual lazy classification scheme.

While there is not a large difference between the knowledge base approaches, there is a large time improvement in comparison to the traditional pattern structures or binary FCA approaches.

### 3.3.6. Other ML models experiments

We trained several machine learning models with the dataset we have to observe how our work measures to common classification models. The experiments were run two times, one with binarized attributes and another with floating-point values of the tf-idf model. The models used include:

- Decision Trees
- Gaussian Naive Bayes

- Support Vector Machines
- Logistic Regression
- Random Forests

The results show a close performance in relation to the pattern structures, which was the SVM model in the case of binary attributes with an F1 value of 0.96 but less time needed to reach a classification, with the best average time in the case of Decision Trees model with floating-point values with 0.008 seconds. The times needed in most of the models used were better than the experiments with the knowledge base.

Some of these models are rules-based, and others are model-based, but in all of these models, there is not a possibility for explaining the results simply. This can be simply done by returning the attribute intersection that led to the classification if the lazy classification algorithm was run, or the classifier in case of the knowledge base experiments.

## 4. Discussion

The results of the experiments show that the best performance in terms of F1 score was given by the binarized attributes. The problem with such methods is their flexibility to perform well when new data is presented.

When pattern structures are used, the less restrictive the values representation and intersection operator, the better it performs. Thus, the best pattern structure was the Min pattern structure, followed by the Max and finally the Interval pattern structure.

The introduction of knowledge bases will save a lot of time searching for the best fit of intersected attributes that could produce a classification, but other questions need to be addressed, like how often the knowledge base must be updated, and how to ensure that there is no bias due to the old data. It is worth noting that building the knowledge base while keeping the floating-point values of the attributes and applying the intersection operator to include (or exclude) ranges of values that might give incorrect results in general, but be locally successful gives better accuracy but needs more time, so a trade-off is established.

While the time needed for common machine learning models is less in most cases, this presents an opportunity for improving the implementation of the knowledge base classification, since the implementation of the used machine learning models (sk-learn) applies multi-threading when possible, which speeds up the classification.

The metrics used to assess such kind of experiments also come to discussion, as the nature of the data and the methods give rise to issues when using F1 for example instead of recall, as in such cases, where the cost of getting Type I errors is high.

## 5. Conclusion

We presented a knowledge base approach for lazy classification using pattern structures of underground forums messages, and the results of the use of the knowledge base are promising in terms of saved time needed to reach a classification.



## 6. Acknowledgements

The article was prepared within the framework of the Basic Research Program at HSE University, RF

## References

- [1] C. P. R. Team, Check point research reports a 38% increase in 2022 global cyberattacks, 2023. URL: <https://blog.checkpoint.com/2023/01/05/38-increase-in-2022-global-cyberattacks/>.
- [2] R. Future, *The Threat Intelligence Handbook: A Practical Guide for Security Teams to Unlocking the Power of Intelligence*, 1st. ed., CyberEdge Group, 2018.
- [3] B. Ganter, R. Wille, *Formal concept analysis: mathematical foundations*, Springer Science & Business Media, 2012.
- [4] S. Ferré, M. Huchard, M. Kaytoue, S. O. Kuznetsov, A. Napoli, *Formal Concept Analysis: From Knowledge Discovery to Knowledge Processing*, Springer International Publishing, Cham, 2020, pp. 411–445. URL: [https://doi.org/10.1007/978-3-030-06167-8\\_13](https://doi.org/10.1007/978-3-030-06167-8_13). doi:10.1007/978-3-030-06167-8\_13.
- [5] M. Kaytoue, S. O. Kuznetsov, A. Napoli, S. Duplessis, Mining gene expression data with pattern structures in formal concept analysis, *Information Sciences* 181 (2011) 1989–2001.
- [6] A. Masyutin, Y. Kashnitsky, S. O. Kuznetsov, Lazy classification with interval pattern structures: Application to credit scoring, in: *FCA4AI@IJCAI*, 2015.
- [7] W. Saidi, Formal concept analysis based association rules extraction (2012) 490–497.
- [8] M. Obitko, V. Snasel, J. Smid, V. Snasel, Ontology design with formal concept analysis., in: *CLA*, volume 128, 2004, pp. 1377–1390.
- [9] G. Jiang, K. Ogasawara, A. Endoh, T. Sakurai, Context-based ontology building support in clinical domains using formal concept analysis, *International journal of medical informatics* 71 (2003) 71–81.
- [10] P. Vilakone, K. Xinchang, D.-S. Park, Movie recommendation system based on users' personal information and movies rated using the method of k-clique and normalized discounted cumulative gain, *Journal of Information Processing Systems* 16 (2020) 494–507.
- [11] B. Ganter, S. O. Kuznetsov, Pattern structures and their projections, in: *International conference on conceptual structures*, Springer, 2001, pp. 129–142.
- [12] S. O. Kuznetsov, Scalable knowledge discovery in complex data with pattern structures, in: *International Conference on Pattern Recognition and Machine Intelligence*, Springer, 2013, pp. 30–39.
- [13] S. O. Kuznetsov, Fitting pattern structures to knowledge discovery in big data, in: *International conference on formal concept analysis*, Springer, 2013, pp. 254–266.
- [14] S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, Pearson, 2021.
- [15] K. E. Wolff, A conceptual view of knowledge bases in rough set theory, in: *International Conference on Rough Sets and Current Trends in Computing*, 2001, p. 220–228.