Stratifying Malware Clusters: A Solution Mapping Paradigm

Pavan R Kashyap[1], Phaneesh R Katti[1], Hrishikesh Bhat P[1], Pranav K Hegde[1], Ethadi Sushma[1], and Prasad Honnavalli B[1]

PES University, Bangalore, Karnataka 560085, India

Abstract. In the current cybersecurity landscape, a multitude of malware strains proliferate, often giving rise to a diverse array of variants. Malware clustering has become a common practice to classify these threats into distinct families and identify their similarities. However, existing approaches often conclude with clustering, overlooking the crucial step of mapping solutions to these clustered malware. Hence, this paper introduces a novel framework for mapping solutions to the outcomes of these clustering models. By doing so, it enables a more practical and efficient response to the ever-evolving threat landscape. When a new malware sample is classified, this framework allows for the exploration of solutions from closely related malware variants within the same cluster or sub-cluster. This approach empowers defenders to select the most fitting countermeasures by harnessing the insights provided by the clustering model and our proposed framework. In essence, this research aims to enhance the synergy between malware clustering models and practical cybersecurity defense. By coupling our framework with these clustering models, we facilitate a more informed and targeted response to emerging malware threats, ultimately bolstering our collective ability to safeguard against them.

Keywords: Malware Clustering · Areas of Influence · Solution Mapping · Malware families · Malware Variants

1 Introduction

In the domain of cybersecurity, a persistent and daunting challenge is the relentless proliferation of malicious software, known as malware. The ability to swiftly and efficiently confront this ever-evolving threat landscape is imperative, with a particular focus on reducing human intervention, which can often impede the reverse engineering process. Moreover, when a malware strain achieves any degree of success, a multitude of variants inevitably emerges, scattered throughout the digital wilderness [1]. Analysing each of these variants individually is a resource-intensive task, leading to potential delays in response and mitigation efforts. In response to these challenges, the practice of clustering has emerged as a powerful technique [2], [3], [11].

Malware clustering, a process that groups related malware variants into clusters based on their similarities, has proven invaluable in classifying and understanding malware families. It illuminates the relative similarity of variants within a given family, allowing analysts to discern their likenesses and deviations. Clustering is the initial step in revealing the inherent relationships among these malware strains, shedding light on their evolutionary pathways and common traits. However, clustering alone provides only a rudimentary understanding of the similarity between two variants in any given cluster [4].

The framework proposed by this paper represents a significant advancement in the field of malware analysis. By introducing the concept of mapping solutions to clusters, our aim is to provide cybersecurity defenders with a power full tool for assessing the threat landscape. The key objective is to establish a standardized, quantitative measure for similarity between peer variants and the head of a malware family. This ensures a rigorous assessment, enabling defenders to gauge relationships within clusters more effectively. The integration of clustering insights with practical mapped solutions equips defenders to efficiently combat the evolving malware menace.

2 Current Problems and Limitations

In the realm of malware clustering, several critical issues and limitations persist, underscoring the need to extend the capabilities of the clustering approach followed universally.

A) **Lack of Solution Mapping**: Traditional malware clustering techniques excel in grouping related malware variants based on their similarities. However, they often fall short in providing practical solutions to the threats identified. Clustering alone offers a basic representation of relationships but lacks the ability to suggest appropriate countermeasures, leaving cybersecurity defenders with a puzzle that demands a more comprehensive solution [4].

B) **Overwhelmingly Variable and resource intensive** One of the foremost challenges stems from the absence of a framework that can offer optimal solutions for similar malware variants. Consequently, the prevalent approach involves the exhaustive exploration of databases, malware analysis reports, and various other sources to identify potential solutions. This resource-intensive process encapsulates the overarching challenges encountered when dealing with these notably variable and similar malware variants.

C) **Limitations of Binary Grouping** The binary grouping approach commonly used in malware clustering, where malware variants are either part of the same family cluster or not, oversimplifies the complex relationships that exist among malware strains [5]. This limitation disregards the gradations of similarity that can exist and does not exploit the full potential of the clustering data.

This paper focuses on addressing these challenges head-on by devising a solution mapping framework to complement the existing malware clustering approaches.

3 Solution Mapping Framework

It is essential to have carried out behavioral-based clustering before following this methodology. There are several proposed approaches [6], [7] to carry out behavioral-based malware clustering.

Our solution mapping framework focuses on mapping a plethora of generic and specific solutions carefully tailored to each malware family cluster. This mapping ensures that appropriate solutions or solution spaces can be parsed effectively when a new malware sample is identified and modelled into one of the given clusters.

General solutions that are mapped to each cluster encapsulate solutions that are shared among all or most samples within that cluster, forming a common solution space. However, general solutions alone may not provide the analyst with adequate guidance for implementing specific techniques to mitigate the particular malware sample at hand. Specific solutions, on the other hand, are tied to individual samples within the cluster, thus contributing to the set of solutions tailored to that specific sample. Therefore, when we refer to specific solutions, it implies solutions that are specific to a given sample within a defined spacial region.

We begin by considering a single malware family cluster. On inspection, cluster samples closely situated to the cluster center exhibit the most characteristic traits of a particular malware family. Conversely, as samples disperse further from the center, their resemblance to the family's behaviors diminishes. Therefore, effective solution mapping within the cluster necessitates not only the identification of the cluster center but also a well-defined stratification of the cluster.

The initial step towards efficient solution mapping involves identifying the shape of the cluster, which serves as the basis for determining the cluster center. In an ideal scenario, the cluster exhibits a circular shape, with the center of the circle serving as the cluster center. For clusters that deviate from a circular form such as an ellipse or an oval, the center is determined based on the shape and size of the cluster.

It's crucial to note that the cluster center represents a hypothetical concept that may not correspond to any specific sample within the cluster. This conceptual center is the most representative entity of the malware family, and it forms the foundation for our approach.

Once the cluster center is established, we introduce a hierarchy of **areas-of-influence** within each cluster. Each circle of influence represents a distinct level of similarity to the hypothetical cluster center. This hierarchical approach is aimed at segregating the solution-mapping into different sections within a given malware family cluster.

The first area of influence, sometimes referred to as the ">=95% cluster" or the "**First-Degree Circle**," includes samples that closely resemble the malware family's core characteristics. This innermost circle maps to only 5% of the overall circle area. For an ideal cluster, the radius of this circle is approximately 0.20 times the cluster radius(R).

The formula to calculate the radii or radii ranges for each area of influence is as follows :

Radius/Radius range for a given area of influence (k_i) =

$$(\sqrt{(1 - UpperboundArea_i\%)} - \sqrt{(1 - LowerboundArea_i\%)}) * R \quad (1)$$

Subsequent areas of influence are structured as follows:
A) The "**Second-Degree Circle**" (samples falling within the 75% to 95% region) comprises samples that closely mirror the core traits of the malware family. Although they may exhibit some unique features, these characteristics align predominantly with the family's core attributes. The second-degree circle in an ideal circular cluster has an internal radius of 0.20R and an external radius of 0.50R.

B) The "**Third-Degree Circle**" (50% to 75% region) includes samples with partial resemblances to the core characteristics of the malware family. The internal radius of the third-degree circle is 0.50R and its external radius of approximately 0.70R for an ideal circular cluster.

C) The "**Fourth-Degree Circle**" (15% to 50% region) encompasses samples that moderately resemble the core family attributes but may also possess specific characteristics significantly distinct from the general family behavior. The fourth degree circle of an ideal circular cluster is mapped onto the cluster between the radius range 0.70-0.92R.

D) The "**Fifth-Degree Circle**" (0% to 20% region) consists of samples located at the outermost periphery, exhibiting limited resemblance to the core family characteristics. These samples exhibit extremely different behaviours and therefore lie in the extreme ends of the cluster.
Figure 1 highlights how an ideal circular cluster containing all the areas of influence mapped onto it, would look like. Each circle of influence resembles the shape of the cluster; if the cluster is an ellipse, all the areas of influence mapped onto it will be elliptical in nature.

This stratification enables the hypothetical assessment of how closely a given sample aligns with the cluster center. Based on their similarity to the center and the area of influence they belong to, suitable solutions can be mapped to each torus-shaped circle of influence, ensuring that each circle contains solutions that are closely related to one another.

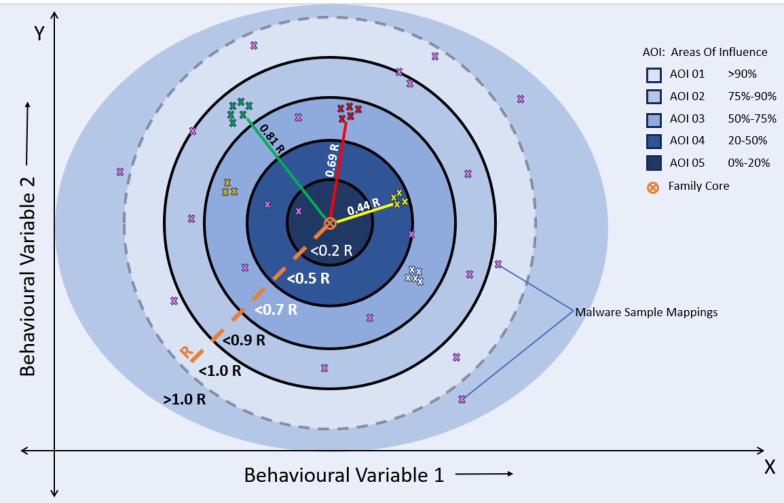

Fig. 1. Malware Family 1's cluster with different areas of influence mapped

Whilst we are one step closer to completing the framework, this initial stratification alone is insufficient. While it accounts for the similarity of samples to the center, it does not consider the similarity of samples to one another within their proximity. This consideration is vital, as samples in close proximity often exhibit very similar behaviors. To address this, we introduce the concept of sub-clustering.

Sub-clustering involves further partitioning each circle of influence into smaller sub-clusters. Each sub-cluster contains samples that are nearly identical to one another within that specific circle of influence. Conventional clustering algorithms, such as K-means[13], can be employed to establish these sub-clusters. The elbow method helps determine the ideal number of entities and clusters within each circle of influence.
Figure 2 displays these sub-clusters within a given circle of influence for the cluster of Malware family X.

It is important to note that these sub-clusters are generated for every circle of influence, except the first-degree circle and the fifth-degree circle. The innermost circle is technically a cluster in itself i.e it represents samples that are all very similar. This is the reason no sub-clusters are generated for the innermost circle. When we claim that a particular malware sample is 0% similar to a certain malware family, then it implies that these samples can lie anywhere outside the inner boundary of the fifth-circle.This suggests that there exists no outer boundary for the fifth-circle in the first place. When there exists no outer bound, sub-clustering does not hold true anymore. Therefore, no sub-clusters are generated for all samples lying in the fifth-degree circle.

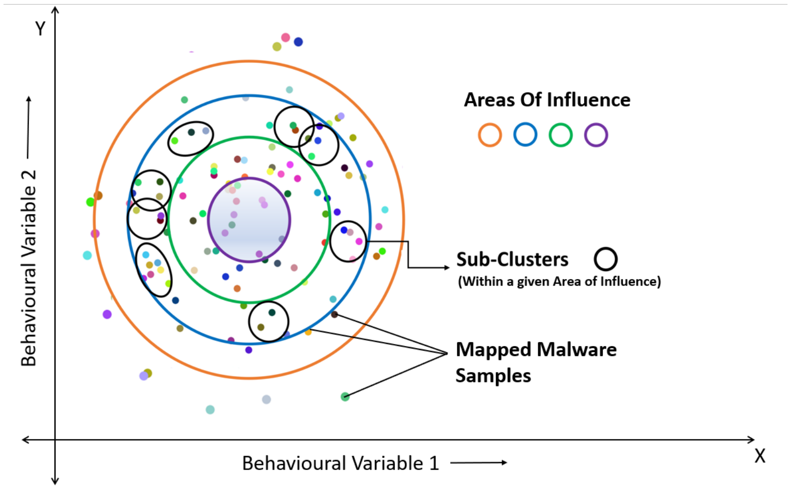

Fig. 2. Malware Family 1's cluster with sub-clusters in respective areas of influence.

Each **torus-shaped circle** of influence (Second degree circle to Fourth degree circle) contains multiple sub-clusters mapped onto it, enhancing the granularity of our approach.

4 Solution Mapping

Once the stratification is established, techniques to place solutions into these fundamental units is undertaken.The solution mapping approach in this paper leverages the use of the "Balanced Tree" data structure, commonly known as **B-Trees**, to represent the entire solution space for a given malware family cluster. This choice of data structure is grounded in two fundamental reasons:

A) **Hierarchical Solution Representation**: The solution space is inherently hierarchical, consisting of both general solutions applicable to a broad spectrum of samples within a cluster and specific solutions tailored to distinct strands of malware within the same cluster. Upon classifying a new malware sample and mapping it to a particular cluster, analysts can efficiently navigate the solution space by first implementing the general solutions linked to the cluster and gradually moving towards the more specific ones. This hierarchical relationship is aptly captured through the structure of a tree, where parent nodes encapsulate more general solutions, and child nodes encapsulate more specific solutions.

B) **Dynamic Adaptability**: The solution space is a dynamic entity, constantly evolving as new solutions and fixes are discovered. Consequently, the structure representing the solution space must adapt accordingly. In the proposed B-Tree framework, the most general solutions find their place toward the root of the tree, while the latest and most specific solutions are appended as leaf nodes. B-Trees inherently maintain a balanced structure, ensuring that new nodes can be added or existing ones deleted without causing structural skew. This dynamic adaptability is essential for preserving the general-to-specific solution pattern while accommodating the ever-changing nature of the cybersecurity landscape. B-Trees perform auto-adjustments, guaranteeing that all leaf nodes remain at the same level within the tree, thus supporting an efficient and responsive framework for managing the evolving solution space.

Each sub-cluster in the malware family is considered a fundamental unit for solution mapping. To appropriately map solutions to each sub-cluster it is essential to identify the core static and dynamic behaviors of the samples in that sub-cluster. To do so, we use YARA rule generators[14], [15], [17] and Behavioral Rule generators[16] to generate YARA and behavioral rules for each sub-cluster. These rules serve two purposes:
A) The rules provide malware analysts with an extensive list of characteristic features (static and dynamic). This can serve as the basis for solution creation.
B) They highlight how different or similar a given malware sub-cluster is to the malware family's first-degree circle. This helps defenders identify how different the solutions must be for the samples in that sub-cluster.

An appropriate B-tree is generated for each sub-cluster with the sub-cluster number being the root node for that B-tree. Within this sub-cluster tree, the parent nodes encapsulate general solutions tailored to the sub-cluster as a whole, while the child nodes house specific solutions relevant to individual samples or more specific groups within the sub-cluster. This hierarchical structure ensures that the specialized knowledge developed for a sub-cluster can be efficiently and effectively applied to all its constituent samples, a pivotal feature for our comprehensive solution mapping framework.

Each B-tree is specific to its sub-cluster i.e it contains solutions addressing the samples only in that cluster. On occasions where samples lie in the intersection of two sub-clusters within a circle of influence, the B-trees of both the sub-clusters can be consulted to devise suitable solutions.

Once all the numbered B-trees are successfully created, they are connected to their appropriate circle of influence number node. All the circle of influence nodes are further connected to the final root node, which is the hypothetical center node. Figure 3 highlights how this tree of tree structure is established.

Some observations from the B-tree highlighted are as follows:
A) There exists a single B-sub-tree under the first degree circle node. This aligns with the idea that the fundamental unit for solution mapping for the innermost circle is the innermost circle itself.

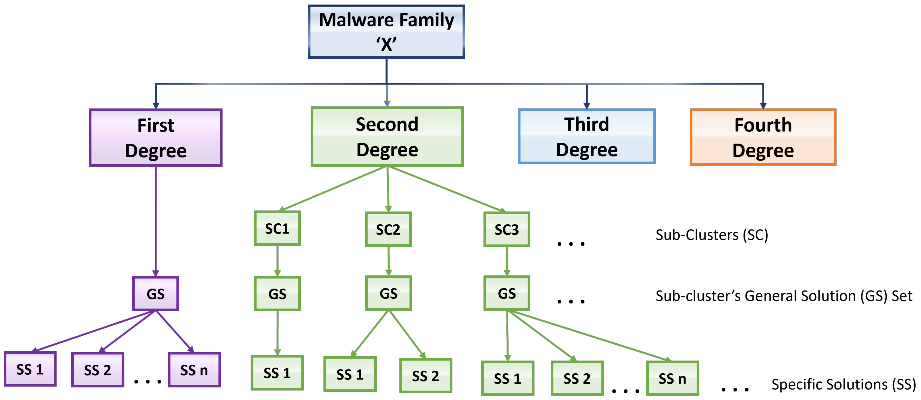

Fig. 3. B-tree for a Malware Family

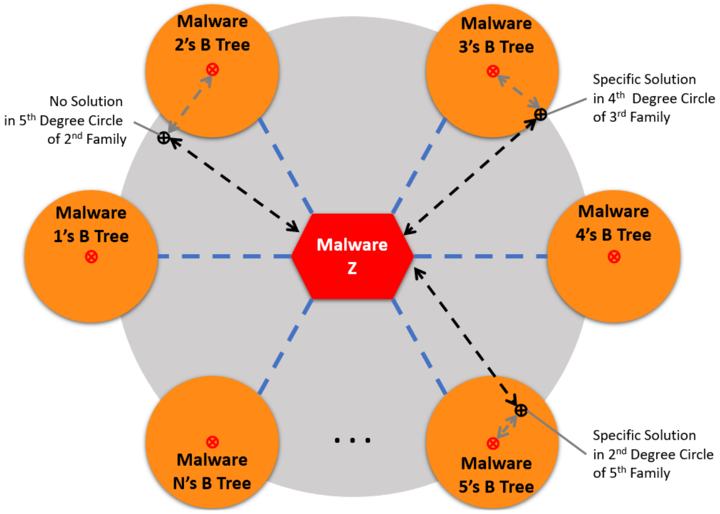
Malware 2's B Tree
Malware 3's B Tree
Specific Solution in 4th Degree Circle of 3rd Family
No Solution in 5th Degree Circle of 2nd Family
Malware 1's B Tree
Malware Z
Malware 4's B Tree
Malware N's B Tree
. . .
Malware 5's B Tree
Specific Solution in 2nd Degree Circle of 5th Family

Fig. 4. Top level view of the Overall B-Tree

B) The B-tree generated represents the solution hierarchy that exists for a given malware family. When a new sample is clustered into this family, its positioning in the circle of influence and sub-cluster can be identified. Parsing those specific sections of the family's B-tree will provide defenders with the effective solutions they need to handle the new sample.

C) Considering that we do not sub-cluster the samples lying in the fifth-degree circles, their solutions do not appear in the final family B-tree. Sub-clustering malware samples that lie in the fifth-degree circle can complicate the mapping of appropriate solutions to each family tree, and therefore they are rightfully ignored.

Each of the family's B-trees can be further connected to generate a large B-tree for all malware samples. The root of this final B-tree can be **Malware Z**, a hypothetical malware sample that encapsulates the behaviors of all malware samples in existence. It serves as the root as it resembles all the malware samples that are existent or will be crafted in the future.Should a particular malware sample resemble multiple families, the solution B-sub-trees of those respective families can be selected and comprehended from Malware Z as seen in figure 4. This nested structure of solution-mapping, allows for efficient management of the complex relationships among malware samples, clusters, and sub-clusters.

5 Framework in Action

In this section, we present a practical demonstration of our proposed framework, offering empirical validation of its real-world utility. We selected a clustering graph image from the research conducted in [11], a study focusing on classifying malware based on their respective families, aligning with the framework's core application.

Our framework's implementation begins with the delineation of areas of influence around the hypothetical center, as depicted in the figure 5. We then proceed to pinpoint sub-clusters within each circle of influence, further enhancing the classification by grouping samples that exhibit comparable behaviors based on their proximity and rules (YARA and behaviour rules) generated, with sub-clustering executed through established algorithms like K-means. While it may appear highly detailed, this process proves to be exceptionally effective in the long run by enhancing the quality of solutions within the future B-tree.

The true potency of our framework becomes evident when a new malware variant emerges. Leveraging the capabilities of existing clustering models [11], [3], this variant is plotted into one of the existing sub-clusters depicted in the figure, within the relevant circle of influence. This results in immediate access to a highly optimized solutions belonging to that specific sub-cluster (by parsing through its B-tree). This approach yields solutions closely aligned with proven effective strategies, as supported by sources such as [12] for XorDdos malware variants and other similar sources for other different malware families mentioned

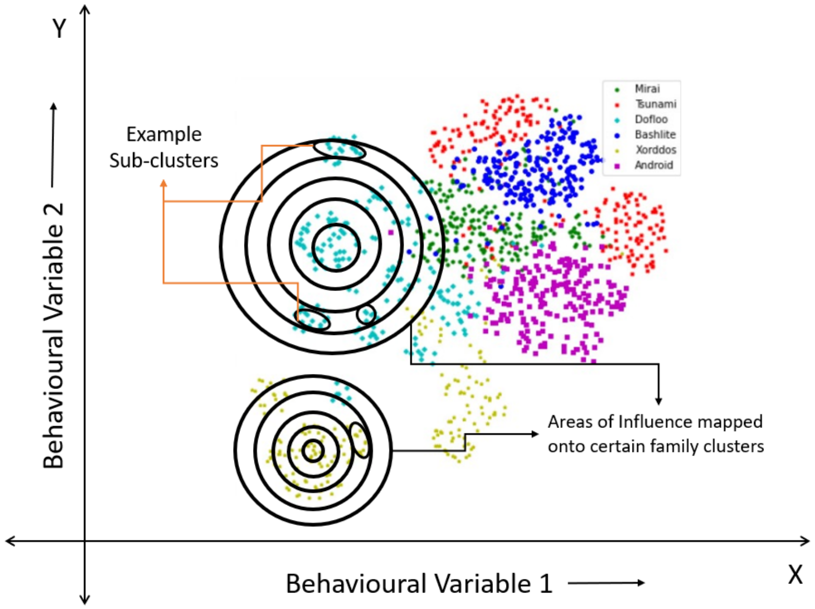

Fig. 5. Cluster and sub-cluster delineation on existing clustering graph

in [11]. The practical application of our framework empowers defenders with an efficient and responsive approach to emerging malware variants, bolstering the overall strength of cybersecurity defenses.

6 Challenges and Limitations

Whilst the approach of generating a forest of B-trees can significantly solve the issues posed by mere clustering, there are some challenges and limitations we face when this approach is used in practice. Some of them are:

A) **Tree traversal skews**: Significant amount of time must be spent to generate this sophisticated B-tree forest (for all the malware families that a security organization deems vulnerable). If new variants of only a particular malware family or family sub-cluster keep recurring, then only solutions associated with that particular B-tree will be used– the rest of the B-tree solutions exists idly without use. If the tree traversal gets skewed and all existing solution spaces are never invoked or used, then it effectively wastes resources (for storage and maintenance of the tree) and time (to generate the entire B-tree).

B) **Heavy Dependence on Clustering Algorithms**: The solution mapping approach is only as effective as the clustering algorithm used to cluster the malware samples into families. Incorrect clustering can lead to inaccurate B-tree generation. Considering how tedious the B-tree generation is, incorrect mapping of samples to malware families can exploit the organization's time and resources significantly.

C) **Overcrowding**: In clustering situations where there is heavy overcrowding, dropping the areas of influence into a malware family cluster to distinguish the similarities becomes tedious. Furthermore, overcrowding of certain sub-clusters and under crowding in some other sub-clusters in the same circle of influence can effectively cause an imbalance in the static and dynamic rules generated for each sub-cluster.

7 Future work

In response to the ever-evolving challenges in the field of cybersecurity, it is imperative to look ahead and shape the future of security solutions. To this end, we identify several critical areas that demand our focus and innovation, especially in the realms of malware clustering and solution mapping, paving the way for enhanced protection against emerging threats. The following key directions for future work have been outlined as part of this endeavor:

A) **Enhanced Clustering Algorithms:** Future endeavors should focus on the refinement and optimization of clustering algorithms for the sake of solution mapping. Developing more sophisticated approaches for grouping malware variants based on behavioral, structural, and genetic similarities will be paramount. The focus of clustering must be on identifying the most distinguishing characteristics of each malware family and mapping them onto the plane in such a way that there is no overcrowding of points in a particular region. These two features will ensure that the clustering algorithms work with greater accuracy and precision. Greater is the clustering accuracy, higher is the possibility of the B-tree forest being accurate.

B) **Dynamic Mapping of Solutions:** As of now, the B-tree has to be manually crafted and filled. Future endeavours can focus on devising an automated architectural framework that maps new solutions gathered via Threat Intelligence into the B-tree at appropriate locations.

C) **New data structures**: Future work can focus on implementing the proposed approach and identifying fundamental flaws with the use of the B-tree data structure. Ideas of using novel data structures to map solutions can be proposed.

D) **Solutions in Fifth-degree circle**: Our current framework excludes solutions for samples that lie in the fifth-degree circle. Novel ideas on how these solutions can also be mapped to some part of the outermost B-tree forest, can be devised.

E) **Cross-Platform Compatibility**: Ensuring that the solution mapping framework is compatible with a wide range of operating systems, network environments, and cybersecurity tools, enabling seamless integration and adaptability.

F) **Scalability**: Investigating methods to scale the solution mapping framework to accommodate the growing volume and complexity of malware samples, while maintaining high performance and accuracy.

8 Conclusion

As the proliferation of malware variants continues unabated, the need for rapid and efficient threat analysis is paramount. Malware clustering has been a crucial technique in classifying and understanding malware families, shedding light on the relationships between variants. However, it falls short in providing actionable solutions.
Our framework addresses this limitation by introducing the concept of areas of influence, which stratify malware variants based on their proximity to the core family traits. This stratification enables the swift identification of optimal solutions when new variants emerge. By pre-constructing a solution B-tree, defenders can efficiently map emerging threats to the most relevant sub-cluster, expediting their solution space.
Through a practical illustration of the framework's capabilities, we have demonstrated the framework's efficacy in rapidly identifying optimized solutions. Ultimately, our framework bridges the gap between malware clustering models and actionable cybersecurity defense. By refining the clustering process, it empowers defenders to efficiently and effectively respond to emerging threats, fortifying our collective ability to safeguard against the evolving menace of malware.

References

1. Cybersecurity and Infrastructure Security Agency, 2021 Top Malware Strains, [online], Last Accessed: 2023, Oct 25, *https://www.cisa.gov/news-events/cybersecurity-advisories/aa22-216a*

2. Faridi, Houtan and Srinivasagopalan, Srivathsan and Verma, Rakesh. (2018). "Performance Evaluation of Features and Clustering Algorithms for Malware". 13-22. 10.1109/ICDMW.2018.00010.

3. Fok Kar Wai and Vrizlynn L. L. Thing, "Clustering based opcode graph generation for malware variant detection", from 2021 18th International Conference on Privacy, Security and Trust (PST), Dec, 2021, IEEE, https://doi.org/10.1109/pst52912.2021.9647814

4. Samanvitha Basole and Mark Stamp, ”Cluster Analysis of Malware Family Relationships”, 2021, 2103.05761, arXiv, cs.CR

5. Pitolli, G., Laurenza, G., Aniello, L. et al. ”MalFamAware: automatic family identification and malware classification through online clustering”. Int. J. Inf. Secur. 20, 371–386 (2021). *https://doi.org/10.1007/s10207-020-00509-4*

6. Lianqiu Xu, Chunyan Zhang, and Ke Tang ”A malware analysis method based on behavioral knowledge graph”, Proc. SPIE 12602, International Conference on Electronic Information Engineering and Computer Science (EIECS 2022), 1260223 (20 April 2023); *https://doi.org/10.1117/12.2668119*

7. Nur Adibah Rosli, Warusia Yassin, Faizal M.A and Siti Rahayu Selamat, “Clustering Analysis for Malware Behavior Detection using Registry Data” International Journal of Advanced Computer Science and Applications(IJACSA), 10(12), 2019. *http://dx.doi.org/10.14569/IJACSA.2019.0101213*

8. Jie Zhang, Senior Director, Engine Development at Fortinet, Published Feb 7, 2018, ”Clustering malware with Machine Learning”, [online], Accessed on: 2023, Oct 25

9. K. Liyanage, R. Pearsall, C. Izurieta, B. M. Whitaker, ”Malware Detection Using Unsupervised Clustering of Binary File Control Flow Graphs”, Malware Detection by CFG Analysis, Dec 2022

10. Pai, S., Troia, F. D., Visaggio, C. A., Austin, T. H., and Stamp, M. (2016). Clustering for malware classification. Journal of Computer Virology and Hacking Techniques, 13(2), 95–107. doi:10.1007/s11416-016-0265-3

11. Chia-Yi Wu, Tao Ban, Shin-Ming Cheng, Takeshi Takahashi, Daisuke Inoue, ”IoT malware classification based on reinterpreted function-call graphs”, Computers & Security, Volume 125, 2023, 103060, ISSN 0167-4048, *https://doi.org/10.1016/j.cose.2022.103060*

12. Microsoft Security, Threat Intelligence, ”Rise in XorDdos: A deeper look at the stealthy DDoS malware targeting Linux devices”, Published: May 19, 2022, [online], Accessed on: Oct 25, 2023, *https://www.microsoft.com/en-us/security/blog/2022/05/19/rise-in-xorddos-a-deeper-look-at-the-stealthy-ddos-malware-targeting-linux-devices/*

13. Pulkit Sharma, Published on: Oct 18, 2023, ”The Ultimate Guide to K-Means Clustering: Definition, Methods and Applications”, [online], Accessed on: 2023, Oct 25, *https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/*

14. VirusTotal, Yara, The pattern matching swiss knife, [online], (Accessed on: 2023, Oct 25) *https://github.com/VirusTotal/yara*

15. Neo23x0, YarGen, Yara rule Generator, [online], (Accessed on: 2023, Oct 25) *https://github.com/Neo23x0/yarGen*

16. Hybrid Analysis, from CloudStrike Falcon, [online] (Accessed on 2023, Oct 25), Available: *https://www.hybrid-analysis.com/*

17. M. Khalid, M. Ismail, M. Hussain and M. Hanif Durad, ”Automatic YARA Rule Generation,” 2020 International Conference on Cyber Warfare and Security (ICCWS), Islamabad, Pakistan, 2020, pp. 1-5, doi: 10.1109/ICCWS48432.2020.9292390.