PassViz: A Visualisation System for Analysing Leaked Passwords*

Sam Parker† Haiyue Yuan‡ Shujun Li§
Institute of Cyber Security for Society (iCSS) & School of Computing, University of Kent, UK

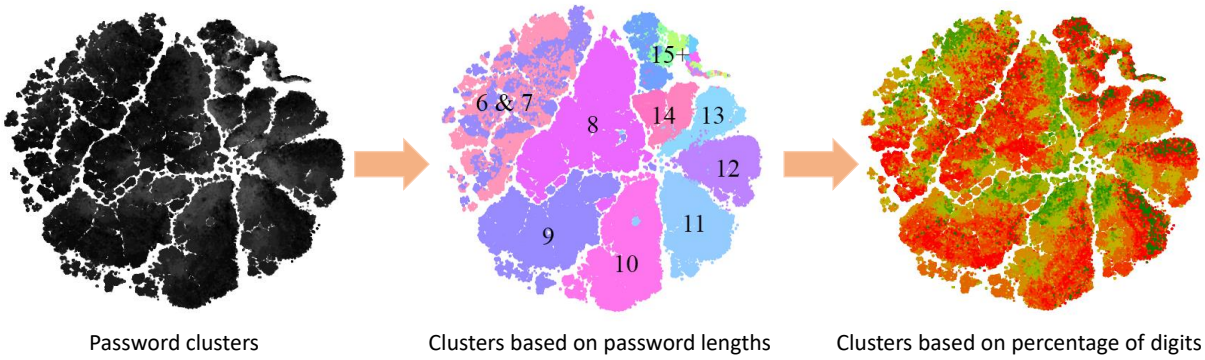

Figure 1: Examples of visualisation of different clusters for 000webhost leaked password database

Abstract

Passwords remain the most widely used form of user authentication, despite advancements in other methods. However, their limitations, such as susceptibility to attacks, especially weak passwords defined by human users, are well-documented. The existence of weak human-defined passwords has led to repeated password leaks from websites, many of which are of large scale. While such password leaks are unfortunate security incidents, they provide security researchers and practitioners with good opportunities to learn valuable insights from such leaked passwords, in order to identify ways to improve password policies and other security controls on passwords. Researchers have proposed different data visualisation techniques to help analyse leaked passwords. However, many approaches rely solely on frequency analysis, with limited exploration of distance-based graphs. This paper reports PassViz, a novel method that combines the edit distance with the t-SNE (t-distributed stochastic neighbour embedding) dimensionality reduction algorithm for visualising and analysing leaked passwords in a 2-D space. We implemented PassViz as an easy-to-use command-line tool for visualising large-scale password databases, and also as a graphical user interface (GUI) to support interactive visual analytics of small password databases. Using the "000webhost" leaked database as an example, we show how PassViz can be used to visually analyse different aspects of leaked passwords and to facilitate the discovery of previously unknown password patterns. Overall, our approach empowers researchers and practitioners to gain valuable insights and improve password security through effective data visualisation and analysis.

Index Terms: Human-centered computing—Visualization—Visualization techniques—Treemaps; Human-centered computing—Visualization—Visualization design and evaluation methods

1 Introduction

Passwords are still the mostly used form of user authentication, especially for websites. Despite ongoing advancements in other forms of user authentication mechanisms, many researchers suggested that the use of passwords would continue to prevail in the foreseeable future [3, 7]. More recently, passwords are often used as part of a multi-factor authentication (MFA) system, where one or more other factors such as "what you have" (token-based) and "who you are" (biometric-based) authentication methods are used to provide enhanced overall security. Despite its wide use, the shortcomings of passwords such as weak passwords defined by human users are well-studied in the research literature [7]. One source of the weak password problem is the conflict of security and usability of passwords: stronger passwords tend to be harder to remember, and easier-to-remember passwords tend to be easier to crack [16, 18]. Human users tend to have different insecure behaviours around password creation, e.g., the mismatch between human users' misperception of a password's strength and its actual strength can lead to creation of weak passwords [1, 18], and many users choose to reuse the same password across multiple accounts [12]. Such weak passwords have led to repeated leakage of passwords from many websites, including some very large-scale incidents. The unfortunate large-scale password leaks give researchers and practitioners opportunities to study such leaked passwords to gain more knowledge and insights about how human users create passwords, in order to find better ways to refine password security controls, e.g., better password policies, password checkers and password management tools.

Most earlier password analysis work was based on simple statistics [11, 13], but data visualisation has been proposed by some researchers to analyse leaked passwords [4, 20, 24], utilising methods such as heat-maps, bar charts, and word clouds. To the best of our knowledge, most past studies on password visualisation are based on frequencies of passwords or segments of passwords, and only a limited number of studies [6, 26] investigated graph-based methods to explore structural relationships between different passwords. Different from existing solutions, this paper presents PassViz,

*This is the authors' version of the accepted paper. Please cite this paper as follows: Sam Parker, Haiyue Yuan and Shujun Li (2023) PassViz: An Interactive Visualisation System for Analysing Leaked Passwords. *Proceedings of the 2023 20th IEEE Symposium on Visualization for Cyber Security* (VizSec 2023), pp. 33-42, IEEE, doi: 10.1109/VizSec60606.2023.00011. For the published version, please visit the publisher's website via the DOI link.

†e-mail: samcparker1999@gmail.com
‡e-mail: h.yuan-221@kent.ac.uk
§e-mail: s.j.li@kent.ac.uk

a new graph-based data visualisation method that leverages edit distances (more precisely Levenshtein distances) and the t-distributed stochastic neighbour embedding (t-SNE) dimensionality reduction algorithm for visualising and analysing leaked passwords in a 2-D space. We implemented PassViz as an easy-to-use command-line tool for visualising large-scale password databases, and also an interactive graphical user interface (GUI) to support interactive visual analytics of small password databases. Using the "000webhost" leaked database as an example, we show how PassViz can be used to analyse different aspects of leaked passwords in a visually meaningful manner and also facilitate the discovery of previously unknown password patterns.

The rest of the paper is organised as follows. Section 2 overviews some related work, followed by a detailed description of the proposed methodology given in Section 3. Section 4 demonstrates different ways of using PassViz to conduct a visual analysis of leaked passwords in the database "000webhost", with a discussion on the limitations of PassViz. The last section concludes this paper with future research directions.

2 Related Work

The understanding of password structures and patterns can provide useful insights into the password creation processes and help develop better password tools such as password strength meters [8]. An early attempt by Morris and Thompson [11] back in the 1970s analysed 3,289 passwords and revealed some basic statistics about passwords structures, where 492 passwords can be identified in open access information sources such as dictionaries and name lists, 86% of passwords can be categorised as one of the 6 classes (e.g., single ASCII character, four alphanumerics, and all lower cases). Similarly, an early work conducted by Riddle et al. [13] investigated 6,226 passwords for a university time-sharing system, and they discovered that user-chosen passwords are commonly based on personal information such as birthday, names or job/project related.

Jakobsson and Dhiman [8] studied the relationship between the percentage of passwords' components such as words, numbers, and other special characters to establish the differences between strong and weak passwords. Differently, Taiabul Haque et al. [17] proposed a hierarchy of password importance that assume that users would mentally classify passwords into different levels based on the perceived importance of different sites (i.e., news portals and banking websites). By observing how users construct passwords following such a hierarchy, they uncovered that unsafe lower-level passwords can be used to crack higher-level passwords due to the behaviour of password reuse with/without modifications. In a study of empirical analysis of large-scale Chinese web passwords, Wang et al. [23] discovered a number of interesting password structures and semantic patterns, which are somewhat different from findings observed in English passwords. They explored 22 types of semantic information such as English names, Pinyin names, date in the format of YYYY, and date in the format of YYMMDD, which contribute to password-cracking strategies.

Leaks of real-world passwords from many websites (e.g., Yahoo, RockYou, and 12306) have become a common phenomenon these days, and they have attracted many researchers' attention to study such leaked passwords in order to gain useful insights about how human users create passwords. One group of methods for facilitating such analyses of leaked passwords is to utilise data visualisation. For instance, Bonneau et al. [2] collected a subset of leaked passwords from RockYou, which contain only 4-digit sequences, and another password database containing only 4-digit PINs to unlock iPhones to study the composition of 4-digit PINs. By visualising the distribution of such PINs using a heat map, they revealed that it is very likely human users choose 4-digit passwords in a format of MMDD (i.e., month-day). They concluded that birthdays have been heavily used as 4-digit passwords.

In another work, Wang et al. [22] conducted a study to compare 4- and 6-digit PINs for English and Chinese users, where heat maps were adopted to visualise date-related features in such PINs. To further explore how dates are used in the password creation process, Veras et al. [21] developed an interactive visualisation tool that combines different visualisation methods including tile maps, radial plots and word clouds. By using the visualisation tool with the RockYou database of over 32 million passwords, they discussed different patterns in passwords including dates, e.g., around 5% of passwords have pure dates, many date-related patterns such as the first days of the month, and holidays were observed. In another follow-up work, Veras et al. [20] conducted qualitative analyses of leaked password databases using semantic grammar to emanate graphical models for visualising high-level dependencies between token classes. Their work captures both syntactic and semantic information, allowing for the identification of regular patterns in passwords that resemble natural language.

Moreover, researchers have been looking at more subtle password patterns that are less obvious for visual observations. Yu and Liao [24] developed a light-weight and web-based visualisation tool combining bar charts, heat maps, tables, and word clouds using the D3 data visualisation library [4] to analyse leaked password databases, which led to the identification of various password patterns (e.g., short and long repeat patterns are common in user passwords, shorter repeating sub-strings are used to form longer repeating sub-strings, and reverse order repetitions are more than forward-order repetitions). In another follow-up work, Yu and Liao [25] developed hierarchical segmentation and optimisation algorithms to visualise and analyse the prefixes and postfixes of human-created passwords.

Apart from date-based patterns in human-created passwords and PINs, keyboard-related patterns have also been investigated by some researchers. Schweitzer et al. [14] discovered that drawing lines connecting the key sequences on a graphical keyboard is not good enough to recognise patterns. Alternatively, they developed a new set of rules using (weighted) arcs and/or loops to help visually recognise keyboard patterns. An analysis based on a large number of human-created passwords revealed that the most common keyboard patterns contained 2-4 continuous keys. Based on this result, Chou et al. [5] used adjacent and parallel keyboard patterns to generate password databases, and subsequently applied them to crack real-world passwords.

To the best of our knowledge, there is limited work that is similar to our work presented in this paper. Shin and Woo [15] attempted to understand password patterns and structure through a data-driven analysis of passwords from four different leaked password databases. They adopted the tensor decomposition method to study password features and identify two dominant features that make a password stronger through similarity distance analysis. Zheng et al. [26] proposed a modification-based approach to explore the spatial structure of passwords in the form of entity-relationship graphs. Similar to our work, they also used Levenshtein distance for comparing passwords. However, their approach differed in terms of utilising the Levenshtein distance to define the edges of vertices in a graph model, while we utilise Levenshtein distances between password pairs to generate distance matrices with subsequent dimensionality reduction for mapping complicated spatial password relationships to a 2-D space for visualisation purposes. Guo et al. [6] also used Levenshtein distances between password pairs to construct a graph showing relationships between passwords, but they used a simple threshold-based approach to define binary connections between passwords, while our work uses a dimensionality reduction method to keep distance between passwords in a 2-D space.

3 METHODOLOGY

The main objective of this work is to develop a tool that facilitates the exploration and analysis of large-scale password databases for researchers and practitioners by leveraging effective data visualisation techniques. To achieve this, we aim to

1. construct high-dimensional representations for passwords in a given database, where passwords with similar structures are positioned close together,
2. embed the high-dimensional representations of all passwords in a 2D space, and
3. develop an easy-to-use toolkit for password visualisation and analysis.

3.1 Quantify similarity between a pair of passwords

The edit distance is a method used to quantify the dissimilarity between two textual strings (e.g., two passwords) by calculating the minimum number of operations needed to transform one string into the other. There are different types of edit distance that involve different sets of editing operations. For instance, Levenshtein distance (hereafter LD) [10] allows three operations: removal, insertion, and substitution of a character in the input strings. Hamming distance takes effect only on passwords that have the same length. In other words, it does not allow insertion or removal. Jaro-Winkler distance is based on the observation that a common mistake when people type is the transposition of two adjacent characters in a string. It favours strings where the first few characters match due to the prefix scale factor in its calculation (e.g., 'password1' and 'password2' have a similarity of 95.6% whereas '1password' and '2password' have a similarity of 92.6%), but passwords come in many formats and do not necessarily have matching prefixes. Moreover, Jaccard similarity and cosine similarity do not account for the order of characters, which can be critical in comparing passwords [2]. Cosine similarity also requires a transformation of the strings into a suitable numerical vector representation, which can complicate the process.

Comparing all the different types of edit distances, we selected LD to quantify the similarity between pairs of passwords. The LD between two passwords is formally defined as the minimum number of single-character edits (insertions, deletions, and substitutions) required to change one password into the other. Mathematically, LD between a pair of passwords a and b can be stated by $\text{lev}_{a,b}(|a|,|b|)$:

$$\text{lev}_{a,b}(i,j) = \min \begin{cases} \text{lev}_{a,b}(i-1,j)+1 \\ \text{lev}_{a,b}(i,j-1)+1 \\ \text{lev}_{a,b}(i-1,j-1)+f\left(a_i,b_j\right) \end{cases}, \quad (1)$$

where $|a|$ and $|b|$ represent the lengths of passwords a and b, respectively, $f(a_i,b_j)$ is an indicator function that equals to 0 when $a_i = b_j$ and to 1 otherwise. The calculation of LD involves a dynamic optimisation algorithm, whose complexity is $O(|a| \times |b|)$.

Table 1: Examples of LDs between three example pairs of passwords

Password 1	Password 2	LD
romans56	blahblah	8
bahamut24ritter	Bonito12	13
rahasia23	abhilash298471	11

3.2 Calculating a distance matrix from all passwords

To facilitate the construction of high-dimensional representations for passwords in a database with respect to other passwords, LDs between all pairs of passwords can be used to create a distance matrix, where each cell represents the similarity between two passwords. In this case, the i-th row and the j-th column represents the LD between the i-th and j-th passwords. Table 2 shows an example of a distance matrix of 10 randomly selected passwords from a leaked password database[1].

Table 2: Examples of a distance matrix of a database with 10 passwords

	anfield	cutlass	denire	GEORGE	21081987	WP2003WP	vjqgfhjkm	hallo123	nathalie	november
anfield	0	7	6	7	8	8	8	7	7	7
cutlass	7	0	7	7	8	8	9	7	6	8
denire	6	7	0	6	8	8	9	8	8	7
GEORGE	7	7	6	0	8	8	9	8	8	8
21081987	8	8	8	8	0	8	9	8	8	8
WP2003WP	8	8	8	8	8	0	9	8	8	8
vjqgfhjkm	8	9	9	9	9	9	0	9	9	9
hallo123	7	7	8	8	8	8	9	0	7	8
nathalie	7	6	8	8	8	8	9	7	0	7
November	7	8	7	8	8	8	9	8	7	0

However, for large leaked password databases, there are too many passwords, so creating a complete distance matrix can incur high time and space complexity. Imagining a best-case scenario for memory usage where we assume that each password is a single character long, amounting to one byte. Even then, a data set with 700,000 passwords would require $700,000 \times 700,000 \times 8 = 3.92$ trillion bits, or 490 gigabytes of memory. Given the high complexity, we resorted to an anchor-based method to make the data visualisation tool more lightweight. The method decided involves selecting a sufficiently small number of representative anchor passwords in the entire database and constructing a distance matrix of all passwords from the anchor passwords only (rather than all). In this way, for a database of size M, we can extract a set of $N \ll M$ anchor passwords and generate an $M \times N$ distance matrix, where each row is an N-d vector indicating how close (LD-wise) each password is to each of the N anchor passwords. This reduced matrix can be easily accommodated in memory and also enables faster computation in subsequent steps. Crucially, this approach still maintains the variance of the data, providing us with a reliable sample for further analysis.

3.3 Dimensionality reduction

In the second step of this process, we used t-SNE [19] to reduce the number of dimensions of each password in the $M \times N$ distance matrix in the previous step from $M > 2$ to just two. As an input to the t-SNE method, the high dimensional distance matrix is passed, and the output is a matrix where each row represents the 2-D coordinates for each password. t-SNE is a machine learning algorithm adept at visualising high-dimensional data and preserving the local structure, enabling it to represent clusters and relationships in the data effectively and making it particularly suitable for our goal of producing a two-dimensional representation. One of its key advantages is its ability to maintain the local structure of the data, meaning it preserves the relationships and clusters that exist in the original high-dimensional data.

[1] https://github.com/danielmiessler/SecLists/blob/master/Passwords/xato-net-10-million-passwords-10000.txt

3.4 Implementation

3.4.1 Python-based command-line tool for visualising large password datasets

We developed a Python command-line tool to process and visualise large password databases as discussed previously in this section[2]. We chose Python due to its extensive range of data and machine learning libraries. The polyleven library package[3] was used for calculating LDs. For the t-SNE algorithm, we used the implementation in the openTSNE pacakge[4]. The visualisation implementation was done with Matplotlib[5] in Python, offering researchers and practitioners various options to interact with the password database to facilitate different follow-up analyses.

3.4.2 Interactive application

In addition, we also developed an interactive web-based application that allows users to explore visualisation of smaller password databases (which can be a smaller subset of a larger password database) in an interactive manner[6]. Here, we offer a glimpse into the capabilities that this interactive application can offer using a smaller password database as an example.

Extraction: Figure 2 shows the extraction functionality of the application, where a set of passwords can be selected and converted into a new graph for closer inspection. It is possible to take this extracted group of passwords and generate a visualisation of them.

Searching: Using regular expressions, a user is able to hide passwords that do not match the regular expression provided. In Figure 3, the regular expression `^[0-9]*$` is used to show only passwords that contain only numbers.

Clustering algorithms: This application has support for performing *k*-means, OPTICS and DBSCAN clustering methods to get a better understanding of where clusters are formed and to allow for easier visualisation of patterns that may not have emerged before. Figure 4 shows the graph after having the OPTICS clustering method performed on it. By performing this clustering algorithm, the application highlights the centre-most passwords within the clusters. This reveals 'andrea' to be the centre of the cluster containing passwords 'andres', 'andrew', 'andrea' and 'andreea'.

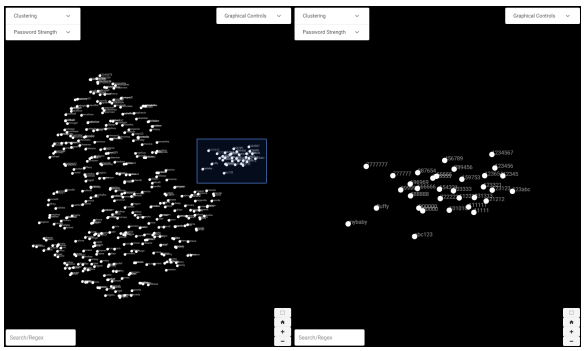

Figure 2: Extracting a group of passwords and opening them in a new window in PassViz

Due to the time and space complexity of processing large password databases from a web browser, this interactive application will be very slow or even impossible to load and process very large password databases. Therefore, we recommend using this interactive application as a complementary tool alongside the command-line

[2] https://github.com/samcparker/passviz-cli
[3] https://ceptord.net/
[4] https://github.com/pavlin-policar/openTSNE
[5] https://matplotlib.org/
[6] https://github.com/samcparker/passviz-gui

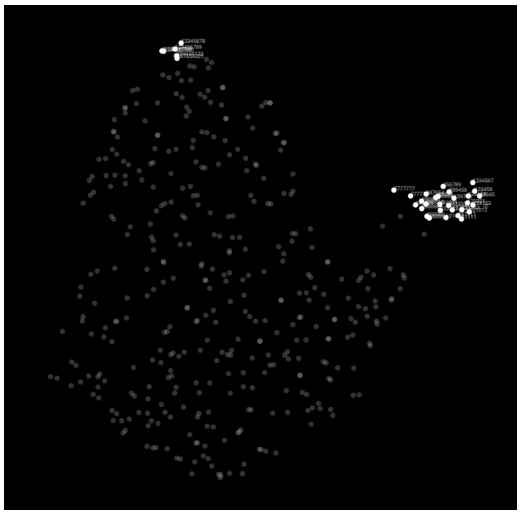

Figure 3: Using regular expressions to highlight individual passwords

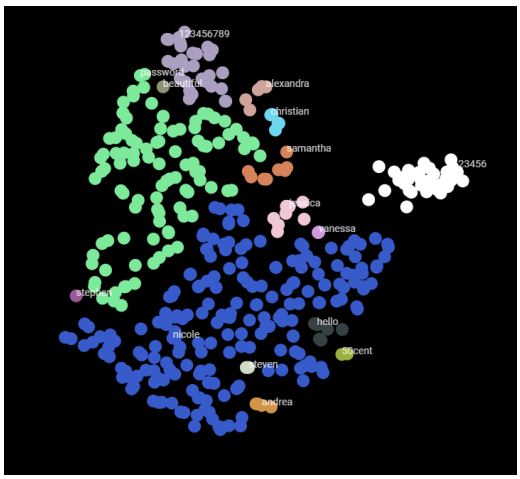

Figure 4: Using OPTICS to cluster the passwords

tool that is better positioned to process and visualise large-scale password databases. This combination enables us to delve into interesting subsets of a large password database to study more hidden patterns, therefore enhancing insights and findings learned from the results from the large database. The interactive application can also be used to test some hypotheses with a small subset of a large password database, and then a more time-consuming process is run using the command-line tool with the full password database. Examples of various password analyses are presented in Section 4 using a leaked database '000webhost'. Nevertheless, one major direction of our future work is to investigate how the time and space complexity of this interactive application can be improved to handle larger password databases, e.g., leveraging parallel processing using multiple cloud servers and GPUs on a single machine.

4 Experimental Results

In this section, we present our work of applying PassViz to analyse passwords in the leaked database '000webhost' to showcase its capabilities and evaluate its effectiveness. 000webhost comprises 15,251,074 clear text passwords, including 720,302 unique passwords. This leaked password database was made public in November 2015, following a security breach from a large web hosting service

000webhost.com. According to a study [20], the origin of the user accounts in 000webhost is reported to be diverse. All accounts in this database are distributed across a wide range of countries, where the largest one (United States) only accounts for 8% of the total population. In addition, the distribution indicates that English passwords are not dominant in the database [20]. By using a subset of randomly selected passwords with the size of 2,000, we were able to construct a distance matrix with the size of $720,302 \times 2,000$. After applying t-SNE dimensionality reduction, we were able to plot all passwords as a 2-D graph and show them in different clusters. As illustrated in Figure 5, PassViz could group all passwords into discernible clusters. To further learn and understand more patterns in this leaked database, more analyses were performed and the findings are presented below.

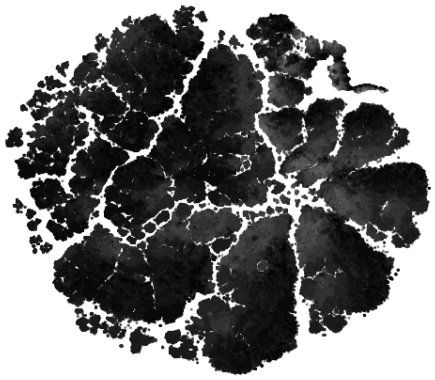

Figure 5: Illustration of clusters of passwords

4.1 Analysis based on password length

As shown in Figure 5, different clusters can be visually observed. However, it is not clear what the most defining factor of the clusters is. By looking into the subsets of the database through the utilisation of the interactive application introduced in Section 3.4.2, these clusters are primarily differentiated by the length of the passwords. We performed a number of analyses to further explore the impact of password length on the visualisation of the password database.

4.1.1 Clusters based on different password lengths

We conducted a subsequent analysis to encode different password lengths with different colours for visualisation. As shown in Figure 6, the visualised database illustrates each password in a colour corresponding to its length. Additionally, a number displayed over each cluster indicates the majority length of the passwords contained within. The size of a cluster corresponds to the number of passwords that have the same length. It is apparent from this method of visualisation that the length of the password is a significant factor in the formation of the clusters, showing that the length of passwords plays a significant role in the structure of passwords within the database.

However, there are exceptions to this observation. One instance is the formation of a mixed cluster, predominantly consisting of passwords of lengths 6 and 7. Despite the minor difference in length, these passwords have enough in common to be grouped into the same cluster. Another exception to this observation is for passwords that have 15 or more characters. Rather than forming individual clusters, these longer passwords combine into a single cluster. The group gradually gets smaller as the length of the passwords increases, reflecting fewer instances of longer passwords in the database. Moreover, it was observed that no cluster contained passwords with fewer than 6 characters, suggesting that 000webhost might have enforced a password-composition policy to have a minimum password length of 6 characters. The identification of larger clusters of passwords with lengths of 8, 9, and 10 is somewhat consistent with the findings reported in [9], which revealed that password-composition policies mandating a minimum of 8 characters typically result in mean password lengths ranging from 9 to 10.

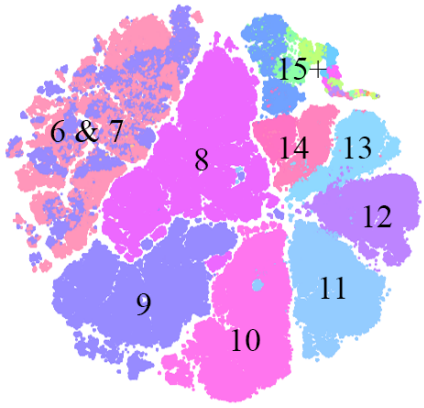

Figure 6: Illustration of colour-coded clusters based on different password lengths

4.1.2 Visualising passwords of the same length

From the previous analysis, it is worth noticing that the defining factor separating leaked password databases into clusters is the length of the password. To further explore, we take passwords of the length of 8 as an example to illustrate how the 000webhost database graph transforms. Around 140,000 passwords are plotted as shown in Figure 7, which gives a better understanding of how graphs are formed, without the aforementioned bias of length down to Levenshtein distance.

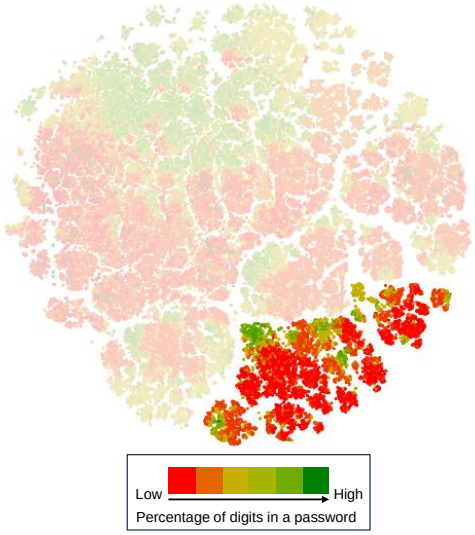

Figure 7: Visualisation of passwords of the length of 8 characters

By visualising the passwords in this way and after further analysis, a common pattern emerged in that many of the passwords in certain clusters had the same character at the same position in each password. In Figure 8, each password has been given a colour based on a property: blue represents passwords where the second letter is 'a',

pink represents passwords where the last letter is '1', and purple represents passwords that abide by both of these properties. It seems that the most defining factor of passwords in our methodology is the position of characters within the passwords. It is striking to see how many users include 'a' as the second letter of their password or '1' as the last letter and is a pattern that would be harder to visualise using more common statistical methods.

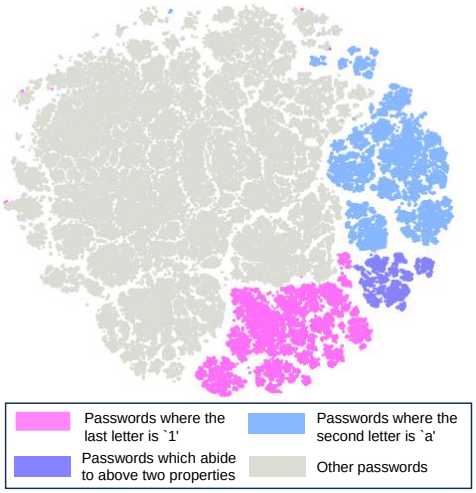

Figure 8: Visualisation of passwords of length 8 that have specific compositions

In addition, looking closely at passwords ending with '1', by isolating the cluster, it appears that many of these passwords only contain a small amount of digits, shown by the majority of passwords appearing red as shown in Figure 7. In comparison, other clusters have more of an orange-to-green hue, showing they contain more numbers.

4.2 Analysis based on the composition of digits in a password

Many existing works have looked into the composition of a password [9]. How digits play a role in creating a password is often of interest to researchers and practitioners. Here, we present a number of analyses utilising PassViz to help derive insights from the large-scale password database by looking at the composition of digits within the password.

4.2.1 Visualisation based on the percentage of digits in a password

Upon assessing the numerical composition of passwords, they were colour-coded based on the percentage of digits in a password, where passwords with dark green colour have the highest percentage of digits and passwords with dark red colour have the lowest percentage of digits. As shown in Figure 9, the visualisation revealed that the utilisation of the numerical composition of a password can separate passwords within their clusters, with one side of the cluster containing passwords having a high percentage of digits, and the other side containing passwords with fewer digits. There appears to be a gradient across all clusters, visualising the change in the number of digits contained within passwords.

To facilitate the analysis, we present Table 3 which displays the distribution of numerical composition in the 000webhost database. The table reveals that 21% of passwords in the database contain 20% digits, while 17% and 16% of passwords have 10% and 30% digits, respectively. Including 5% of passwords that have no digits, 59& of passwords in the 000webhost database have less than 30% numerical content, which makes the overall graph lean towards the colour of red.

Table 3: Distribution of the percentage of digits in each password in the 000webhost database

% digits	% passwords	% digits cont.	% passwords cont.
0	5	60	7
10	17	70	4
20	21	80	4
30	16	90	0.7
40	10	100	0.06
50	13	-	-

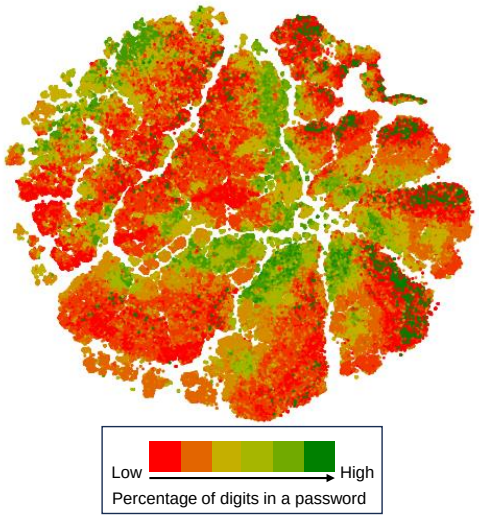

Figure 9: Visualisation of passwords that have different percentages of numbers

4.2.2 Visualisation based on the dominating position of digits

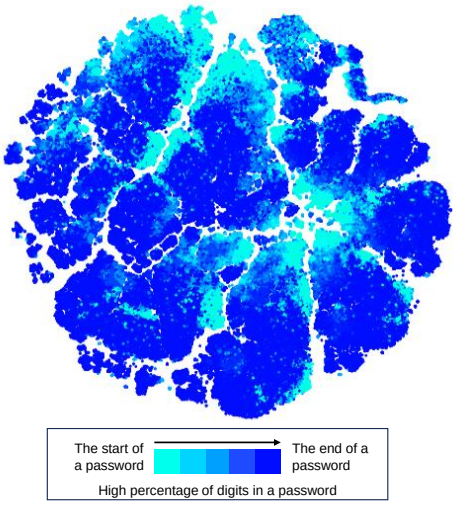

Figure 10: Visualisation of positional distribution of digits within a password

An interesting property to be examined is the positional distribution of digits within a password. This metric measures the location of digits within a password, where values near 0 signify that numbers are primarily concentrated towards the beginning of the password, and values closer to 1 indicate numerical characters mostly towards the end. A distribution ratio around 0.5 suggests either an even dispersion of digits or a lack of digits altogether. Figure 10 depicts this metric, with the light blue shaded passwords indicating a higher quantity of digits towards the start of the passwords and the dark blue passwords signifying a greater presence of digits towards the end. A noteworthy observation is the comparative scarcity of passwords with a high predominance of digits towards the start as opposed to passwords with a majority of digits towards the end. This implies a bias towards appending numbers at the end of the passwords.

4.3 Analysis based on specific requirements

To assist the exploration of a large-scale password database to provide more insights, PassViz has the capability and flexibility to produce visualisation based on more specific requirements. Here we present some examples of utilising PassViz to learn password patterns and structures.

4.3.1 Visualisation based on a given string

All instances containing the word 'hello' in the 000webhost database were highlighted as shown in Figure 11. These instances are relatively scarce, however, there are occasionally groupings. This suggests that while the strings may appear similar based on their contents, it is not a decisive factor in global cluster formation, only in local formation within clusters.

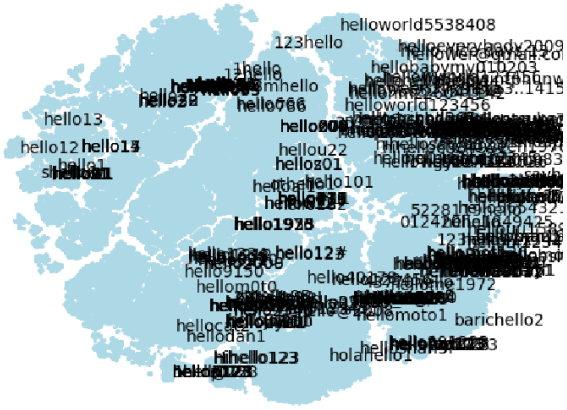

Figure 11: Visualisation of passwords that contain the word 'hello'

4.3.2 Visualisation based on passwords containing years

In addition, existing works discovered that numbers that represent dates/years have been frequently used in the process of password/PIN creation [21, 22]. We are interested in visualisation the distribution of such information in the 000webhost database using PassViz to see if there are any interesting patterns that can be discovered. As depicted in Figure 12, passwords containing dates from the years 2000-2099 are highlighted in blue, such as 'amado2009', while those containing dates from 1900-1999, like 'small1970sman', are marked in red. These dates were chosen specifically as they have the most relevance to current users.

In this visualisation, it can be seen that a small portion of passwords containing these year-related numbers are scattered across the graph. The larger clusters in blue and red formed primarily consist of passwords where the year forms the end part of the password, like 'amado2009', suggesting that a substantial portion of users with a date in their password append a specific year to a base word, rather than at the start or in the middle of the password, contributing to the formation of these clusters. On the other hand, the individual points are scattered throughout the clusters representing the less common instances where the year appears in the middle of a password such as 'small1970sman', rather than at the end of the password.

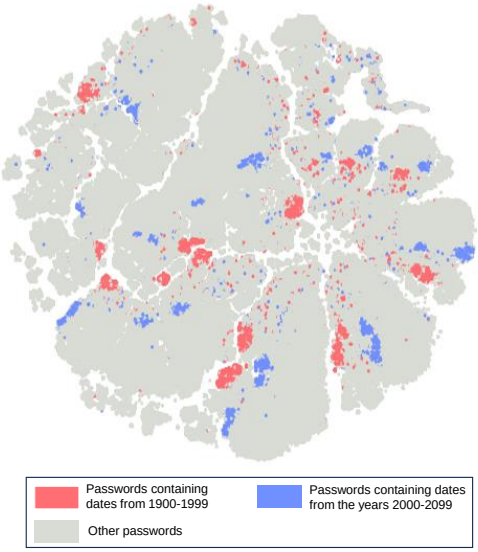

Figure 12: Visualisation of passwords that contain years

4.4 Comparative analysis

4.4.1 Comparing 000webhost with phpbb based on the percentage of digits

This generation methodology in this research can be extended and applied to different databases. To illustrate this, we present the graph for the leaked database 'phpbb'[7] of over 184,000 unique passwords, shown in Figure 13. This shows the percentage of digits in passwords, with red indicating passwords containing no digits, green indicating passwords containing only digits, and colours in-between showing a colour in-between showing the proportion of digits in the password.

Comparing this with Figure 9, an intriguing pattern becomes evident. The proportion of passwords in the phpbb database containing mostly digits is notably larger compared to the 000webhost database shown in Figure 9. Additionally, the significant amount of green highlights the prevalence of passwords composed exclusively of digits.

This gives us an insight into the general security of passwords in each database. The graph generated using the 000webhost database does not show the intense red or intense green that the phpbb database shows, indicating that the passwords used within it are more secure. This could be down to security restrictions imposed on users requiring them to use digits in their passwords. On the other hand, the phpbb database generates a predominantly red and green graph, with only a small amount of colour in between. This indicates that the security restrictions imposed on users were not the same as intense as on 000webhost.

[7] https://github.com/danielmiessler/SecLists/blob/master/Passwords/Leaked-Databases/phpbb.txt

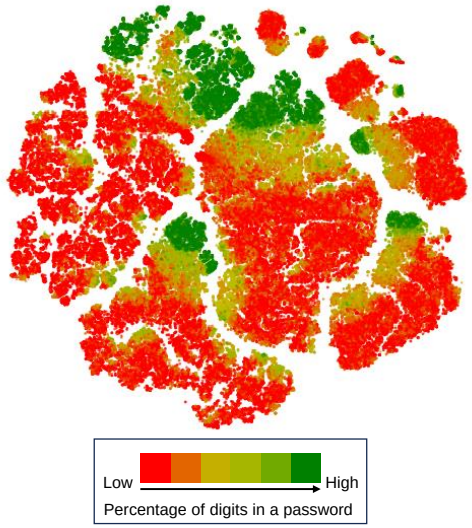

Figure 13: Visualisation of passwords that have different percentages of digits in the phpbb database

4.4.2 Comparing 000webhost with phpbb based on sequences

In this section, we focus on the prevalence of numeric sequences and keyboard patterns in the passwords in each database. For this analysis, a visual representation was generated where the passwords containing numeric sequences (such as '123' and '1234') are marked in red and those satisfying keyboard passwords (consecutive keyboard entries, such as 'qwerty' and 'zxcvb') are marked in blue.

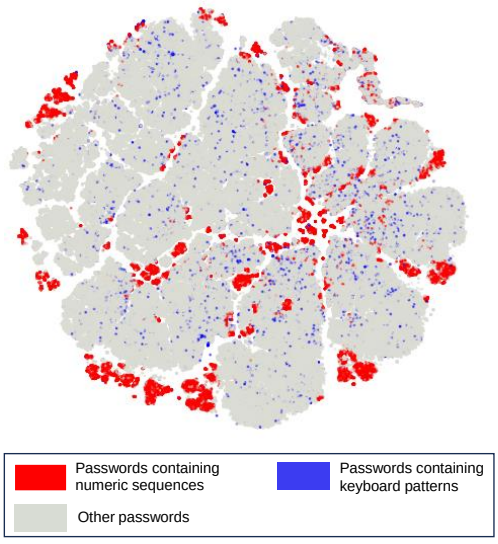

Figure 14: Sequences highlighted in the 000webhost database

The 000webhost database illustrated a pronounced prevalence of numeric sequence patterns in password creation, as shown in Figure 14. A substantial proportion of the passwords contained easily identifiable sequences which start with '123' and increase incrementally. This '123' pattern, despite being a weak password strategy, is in much use among passwords in the 000webhost database.

On the contrary, the phpbb database demonstrated significantly fewer instances of numeric sequence patterns, as shown in Figure 15. This disparity suggests that phpbb users may have had a better understanding of secure passwords, or that the platform itself may have enforced stricter password policies. However, when comparing it with Figure 13, we can see that few passwords in the phpbb database contain digits, which is likely the reason why the '123' pattern is not common inside of the phpbb database.

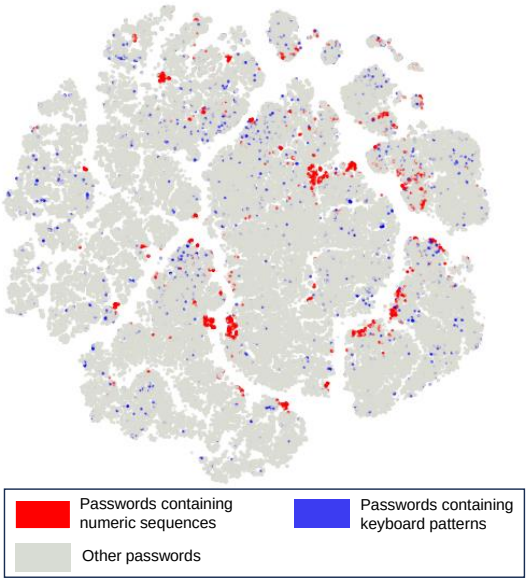

Figure 15: Sequences highlighted in the phpbb database

4.4.3 Intersection between 000webhost and phpbb

Calculating the intersection between two password databases may indicate the similarities between them and will show common passwords between the two. Figure 16 visualises passwords in the phpbb database, with passwords marked in red that also appear within the 000webhost database. The number of intersections between the two databases is 6,091 – 0.84% of 000webhost and 3.3% of phpbb – showing a small amount of commonality between the two databases. This is an important area to look at, as it shows where passwords are being re-used between the two databases and will highlight instances where users are re-using seemingly unique passwords across multiple platforms.

In the figure, it can be seen that intersecting passwords are distributed in a non-uniform manner. There are some areas in the graph where marked passwords have a higher concentration, highlighting that there are certain types of passwords that are more likely to be re-used. After further examination on these groups of passwords, it can be seen that these concentrated areas are formed of passwords ending with the characters of '123'.

Some notable instances are longer passwords that may appear random. By performing this method of visualisation, we can see that they are re-used across both databases, despite appearing seemingly unique. These passwords typically appear at longer lengths and include 'richmond1969', 'serkan737526' and 'nikita040683'. One could expect that these come from the same user having created accounts on both platforms.

There are other longer passwords that appear in both databases, however, these do not appear to be as random and unique as the previous one. These come in the form '1q2w3e4r5t6y', 'abc123def456' and 'qwe123asd456'. Despite being long, these passwords are not necessarily unique and are formed of common keyboard patterns.

Thus, using a long set of characters does not necessarily imply a unique password.

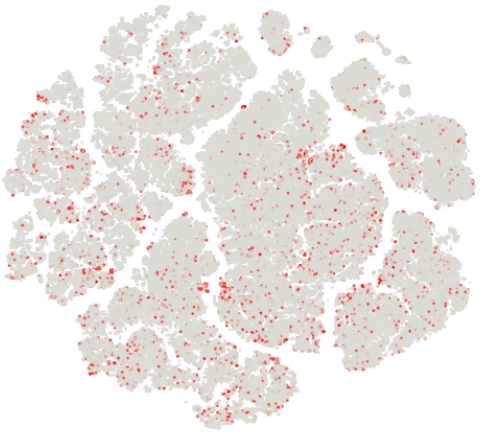

Figure 16: Passwords in the phpbb database, with red dots representing passwords shared with the 000webhost databbase

To get a better understanding of the intersection between these two databases, Figure 17 visualises the graph generated by the intersection of the 000webhost and phpbb databases. As shown in Figure 14, passwords that contain a numeric sequence are highlighted in red, and passwords that contain keyboard sequences are highlighted in blue. This graph reinforces how many passwords reused between the two databases contain numeric sequences, as this is one of the most defining features of this generated graph. Other notable areas are the reuse of passwords containing other keyboard sequences, like 'qwerty', which also appear frequently throughout the graph.

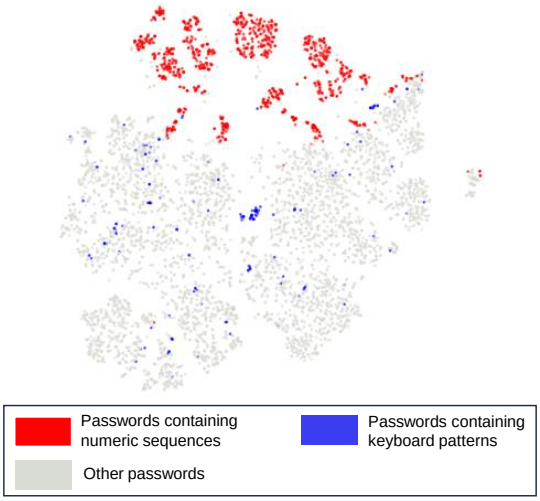

Figure 17: Intersection between the 000webhost and phpbb databases

5 Further Discussions

5.1 Typical use cases

In this subsection, we list some typical use cases of PassViz in real-world applications, based on the examples explained in the previous section. Note that the list is not exhaustive.

- Use Case 1: As presented in Sections 4.1 and 4.2, PassViz can be used to reinforce the comprehension of password structures and patterns, thereby extracting valuable insights about human users' password creation processes. In turn, this will aid in the refinement and development of password tools like password strength meters and password policies.
- Use Case 2: As demonstrated in Section 4.3, PassViz provides flexible ways to allow researchers and practitioners to interact with a large password database with ease to explore finer structures related to subsets.
- Use Case 3: As illustrated in Section 4.4, PassViz allows the comparison between two password databases, in order to reveal cross-database/website patterns that cannot be revealed by studying the multiple password databases separately.
- Use Case 4: All the analyses supported by PassViz can help unveil different aspects of human behaviours in the password creation process, e.g., how they use numbers and keyboard patterns, how they apply character transformation rules to make a password more complicated, and how they reuse or change behaviours across different websites. Such insights related to human behaviours can be useful for a wide range of applications, including development of better tools and also better ways to educate users about password security.

5.2 Limitations

Despite their utility, the generated visualisations do show certain limitations. We discuss some of such limitations below.

One limitation is that some passwords with a small LD could be mis-clustered. For instance, passwords such as 'hello123' and 'hello12', while notably similar with a LD of 1, are segregated as shown in Figure 18. It is likely due to bias towards strings of identical length within a distance matrix. Unavoidable errors introduced by the dimensionality reduction algorithm may be another source.

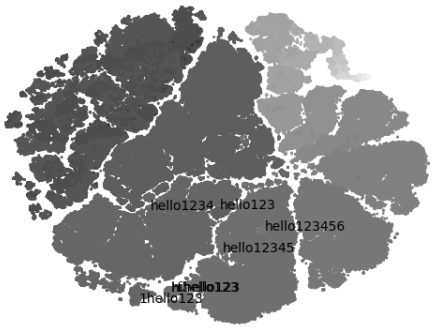

Figure 18: An example of one observed limitation of PassViz for 'hello123' and 'hello12'

A second limitation is that the LD we used is unable to capture all aspects of semantic similarity of two passwords. For instance, 'hello123' and '123hello', despite their perceptible semantic similarity, appear substantially distanced from each other as illustrated in Figure 19. Considering their LD is indeed relatively large (5), the separation can be conceptually explained by the limitation of LD as an edit distance without considering reversing the whole string as a single editing step.

A third limitation is that the use of a dimensional reduction algorithm will unavoidably lead to loss of information for some password pairs (so their distances can be more distorted than others). How to address this is non-trivial since we have to visualise passwords in a low-dimensional (2-D or 3-D) space.

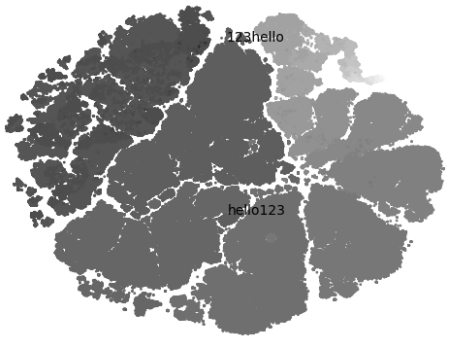

Figure 19: An example of a second observed limitation of PassViz for 'hello123' and '123hello'

5.3 Future Work

We have identified a range of future work directions as described below.

More comprehensive testing and password analyses: The password analyses we conducted and reported are relatively ad hoc, and we only tested PassViz with a number of leaked password databases and some patterns. It will be useful to conduct a more comprehensive analysis with more leaked password databases and a more comprehensive set of patterns. It will also be helpful to design ways to reveal more unknown patterns about passwords. Such further analyses can also involve recruitment of human participants to more confidently confirm the usefulness of PassViz.

Refinement our methodology: The current study uses a distance matrix based on LD for password position generation. While this approach has its merits, it may not necessarily provide the most comprehensive or meaningful results. Future work could focus on exploring alternative distances, dimensional reduction and clustering methods, such as term frequency *n*-grams and other semantic analysis based vectorisation methods, which could potentially show patterns in passwords that might be overlooked or mis-handled by LD. One interesting approach is to use a large language model (LLM) to define a more semantically aware distance and use the LLM to select more representative anchor passwords of the whole password space. Incorporating a password strength meter in the distance metric may also be useful. We also plan to enhance the reconfigurability of PassViz to support different distance metrics and different dimensional reduction methods.

Further development of visualisation tool: An additional practical implication of this research is the further development of a more comprehensive application that implements more features required to visualise and analyse leaked passwords, especially allowing interactive visual analytics of large password databases.

Integration of chatbots: One prospect for the future would be the integration of a chatbot capable of interpreting both graphical input and creating command-line actions. This tool could generate, visualise and analyse graphs autonomously, streamlining the process and reducing the manual workload. With advancements in natural language processing especially LLMs, the development and implementation of such chatbots has become increasingly feasible.

6 Conclusion

In conclusion, the work we have produced has provided visual insights into the underlying passwords and structures within large-scale password databases. It has shown a multitude of patterns and correlations that might have remained hidden in traditional statistical analyses. These visualisations lie not only in their ability to summarise complex databases, but also have the potential to inform password security policies and user education efforts.

References

[1] S. I. Alqahtani, S. Li, H. Yuan, and P. Rusconi. Human-generated and machine-generated ratings of password strength: What do users trust more? *EAI Endorsed Transactions on Security and Safety*, 6(21):e1:1–e1:15, 2019. doi: 10.4108/eai.13-7-2018.162797

[2] J. Bonneau, S. Preibusch, and R. Anderson. A birthday present every eleven wallets? the security of customer-chosen banking PINs. In *Financial Cryptography and Data Security: 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers*, vol. 7397 of *Lecture Notes in Computer Science*, pp. 25–40. Springer, 2012. doi: 10.1007/978-3-642-32946-3_3

[3] L. Bošnjak and B. Brumen. Rejecting the death of passwords: Advice for the future. *Computer Science and Information Systems*, 16(1):313–332, 2019. doi: 10.2298/CSIS180328016B

[4] M. Bostock, V. Ogievetsky, and J. Heer. $\mathbb{D}^3$: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011. doi: 10.1109/TVCG.2011.185

[5] H.-C. Chou, H.-C. Lee, C.-W. Hsueh, and F.-P. Lai. Password cracking based on special keyboard patterns. *International Journal of Innovative Computing, Information and Control*, 8(1):387–402, 2012.

[6] X. Guo, H. Chen, X. Liu, X. Xu, and Z. Chen. The scale-free network of passwords: Visualization and estimation of empirical passwords. arXiv:1511.08324 [cs.CR], 2015. doi: 10.48550/arXiv.1511.08324

[7] C. Herley and P. Van Oorschot. A research agenda acknowledging the persistence of passwords. *IEEE Security & Privacy*, 10(1):28–36, 2012. doi: 10.1109/MSP.2011.150

[8] M. Jakobsson and M. Dhiman. The benefits of understanding passwords. In *Proceedings of the 7th USENIX Workshop on Hot Topics in Security*, pp. 13:1–13:10. USENIX Association, 2012.

[9] S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman. Of passwords and people: Measuring the effect of password-composition policies. In *Proceedings of the 2011 SIGCHI Conference on Human Factors in Computing Systems*, pp. 2595–2604. ACM, 2011. doi: 10.1145/1978942.1979321

[10] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*, vol. 10, pp. 707–710. Nauka, 1966.

[11] R. Morris and K. Thompson. Password security: A case history. *Communications of the ACM*, 22(11):594–597, 1979. doi: 10.1145/359168.359172

[12] S. Pearman, J. Thomas, P. E. Naeini, H. Habib, L. Bauer, N. Christin, L. F. Cranor, S. Egelman, and A. Forget. Let's go in for a closer look: Observing passwords in their natural habitat. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 295–310. ACM, 2017. doi: 10.1145/3133956.3133973

[13] B. L. Riddle, M. S. Miron, and J. A. Semo. Passwords in use in a university timesharing environment. *Computers & Security*, 8(7):569–579, 1989. doi: 10.1016/0167-4048(89)90049-7

[14] D. Schweitzer, J. Boleng, C. Hughes, and L. Murphy. Visualizing keyboard pattern passwords. In *Proceedings of the 2009 6th International Workshop on Visualization for Cyber Security*, pp. 69–73. IEEE, 2009. doi: 10.1109/VIZSEC.2009.5375544

[15] Y. Shin and S. S. Woo. PasswordTensor: Analyzing and explaining password strength using tensor decomposition. *Computers & Security*, 116:102634:1–102634:17, 2022. doi: 10.1016/j.cose.2022.102634

[16] E. Stobert and R. Biddle. The password life cycle: User behaviour in managing passwords. In *Proceedings of 10th Symposium On Usable Privacy and Security*, pp. 243–255. USENIX Association, 2014.

[17] S. M. Taiabul Haque, M. Wright, and S. Scielzo. Hierarchy of users' web passwords: Perceptions, practices and susceptibilities. *International Journal of Human-Computer Studies*, 72(12):860–874, 2014. doi: 10.1016/j.ijhcs.2014.07.007

[18] B. Ur, J. Bees, S. M. Segreti, L. Bauer, N. Christin, and L. F. Cranor. Do users' perceptions of password security match reality? In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 3748–3760. ACM, 2016. doi: 10.1145/2858036.2858546

[19] L. van der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[20] R. Veras, C. Collins, and J. Thorpe. A large-scale analysis of the semantic password model and linguistic patterns in passwords. *ACM Transactions on Privacy and Security*, 24(3):20:1–20:21, 2021. doi: 10.1145/3448608

[21] R. Veras, J. Thorpe, and C. Collins. Visualizing semantics in passwords: The role of dates. In *Proceedings of the 9th International Symposium on Visualization for Cyber Security*, pp. 88–95. ACM, 2012. doi: 10.1145/2379690.2379702

[22] D. Wang, Q. Gu, X. Huang, and P. Wang. Understanding human-chosen PINs: Characteristics, distribution and security. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 372–385. ACM, 2017. doi: 10.1145/3052973.3053031

[23] D. Wang, P. Wang, D. He, and Y. Tian. Birthday, name and bifacial-security: Understanding passwords of Chinese web users. In *Proceedings of the 28th USENIX Security Symposium*, pp. 1537–1555. USENIX Association, 2019.

[24] X. Yu and Q. Liao. User password repetitive patterns analysis and visualization. *Information & Computer Security*, 2016. doi: 10.1108/ICS-06-2015-0026

[25] X. Yu and Q. Liao. Understanding user passwords through password prefix and postfix (P3) graph analysis and visualization. *International Journal of Information Security*, 18:647–663, 2019. doi: 10.1007/s10207-019-00432-3

[26] Z. Zheng, H. Cheng, Z. Zhang, Y. Zhao, and P. Wang. An alternative method for understanding user-chosen passwords. *Security and Communication Networks*, 2018:1–12, 2018. doi: 10.1155/2018/6160125