technologies

MDPI

Article

Multistage Malware Detection Method for Backup Systems

Pavel Novak †, Vaclav Oujezsky *,†, Patrik Kaura, Tomas Horvath and Martin Holik

Faculty of Informatics, Masaryk University, Botanicka 68a, 602 00 Brno, Czech Republic; novakpav@mail.muni.cz (P.N.); 250979@muni.cz (P.K.); tomas.horvath@mail.muni.cz (T.H.); martin.holik@mail.muni.cz (M.H.)

* Correspondence: oujezsky@fi.muni.cz

† These authors contributed equally to this work.

Abstract: This paper proposes an innovative solution to address the challenge of detecting latent malware in backup systems. The proposed detection system utilizes a multifaceted approach that combines similarity analysis with machine learning algorithms to improve malware detection. The results demonstrate the potential of advanced similarity search techniques, powered by the Faiss model, in strengthening malware discovery within system backups and network traffic. Implementing these techniques will lead to more resilient cybersecurity practices, protecting essential systems from hidden malware threats. This paper's findings underscore the potential of advanced similarity search techniques to enhance malware discovery in system backups and network traffic, and the implications of implementing these techniques include more resilient cybersecurity practices and protecting essential systems from malicious threats hidden within backup archives and network data. The integration of AI methods improves the system's efficiency and speed, making the proposed system more practical for real-world cybersecurity. This paper's contribution is a novel and comprehensive solution designed to detect latent malware in backups, preventing the backup of compromised systems. The system comprises multiple analytical components, including a system file change detector, an agent to monitor network traffic, and a firewall, all integrated into a central decision-making unit. The current progress of the research and future steps are discussed, highlighting the contributions of this project and potential enhancements to improve cybersecurity practices.

Keywords: backup; detection; hashes; malware; model; machine learning; system

Citation: Novak, P.; Oujezsky, V.; Kaura, P.; Horvath, T.; Holik, M. Multistage Malware Detection Method for Backup Systems. *Technologies* **2024**, *12*, 23. https://doi.org/10.3390/technologies12020023

Academic Editor: Pedro Antonio Gutiérrez

Received: 16 November 2023
Revised: 29 January 2024
Accepted: 31 January 2024
Published: 5 February 2024

1. Introduction

In today's digital age, in which the vast majority of our activities rely on digital systems and the data contained within them, the attractiveness of these systems to criminals is on the rise. These systems often contain a substantial amount of critical data, and their functionality and constant availability are key requirements. To prevent data loss, and more generally, the loss of these entire systems, one of the most widespread measures currently employed is using a backup system. The backing up of such systems is usually performed regularly, keeping the history of the most recent few backups. In the event of data loss in the system, whether due to malware infection or, more specifically, ransomware, or other failures, data can be recovered, minimizing the loss for the business.

However, given the volume of data, it is usually not feasible to maintain an extensive backup history. Attackers are well aware of this, and increasingly compromised systems are seen with so-called dormant or latent malware. Such malware does not detonate immediately after infecting a system but instead after a certain time delay, which can range from hours to days, or even months. This technique has been used by malware developers for quite some time, primarily to hinder dynamic analysis of samples, which is what most endpoint detection and antimalware systems currently implement. Now, it is also finding applications in ransomware. This technique effectively bypasses the protection offered by backup systems. If you backup an already compromised system with latent malware,

such a backup becomes useless. After ransomware activates, companies often find out by surprise that their supposedly fail-proof backup system is useless because no backup can be used and all backups are compromised. Since backups are one of the major techniques for preventing ransomware attacks, latent malware poses a significant threat. With the increasing complexity of malware threats and, more specifically, ransomware [1], there is a pressing need for more robust and efficient detection techniques.

The detection of latent malware is quite challenging. Typically, behavioral analysis commonly used by commercially available antivirus products does not cover all potential risks. In addition, methods based on known signatures should also be revised for new and unknown samples.

Our research contributes greatly to this effort by introducing a novel and comprehensive solution designed to detect such malware in backups and thus prevent the backup of an already infected image. The approach we take to address this problem consists of multiple components that attempt to recognize the presence of malware, even when its activity may be currently negligible or nearly non-existent, without any outward manifestations.

Our proposed detection system [2] harnesses the capabilities of the Faiss [3] model to compare file hashes and incoming network traffic signatures more effectively. This paper pioneers a multifaceted approach that combines similarity analysis with machine learning algorithms to improve malware detection.

A more complex detection system can discover malware even when latent, and the traditional antimalware detection methods mentioned above fail to do so. Due to the substantial volume of data and the high evaluation speed requirements, we have successfully incorporated Artificial Intelligence (AI) methods. This has resulted in significant time savings, enabling a more efficient practical use of the system. The findings underscore the potential of advanced similarity search techniques to enhance malware discovery in system backups and network traffic. Implementing these techniques will lead to more resilient cybersecurity practices, protecting essential systems from malicious threats hidden within backup archives and network data. A summary of our motivation for this research is as follows:

- The need for robust detection techniques: As malware, particularly ransomware, becomes more sophisticated, there is a pressing need for advanced and efficient detection techniques that go beyond traditional methods.
- Limitations of current detection approaches: The behavioral analysis and signature-based methods employed by antivirus products have limitations in covering all potential risks. Revision is required to deal with new and unknown samples.
- Contribution of research: This research introduces a comprehensive solution designed to detect latent malware in backups, preventing the backup of compromised systems. The approach incorporates multiple components and uses advanced techniques, including AI, for improved malware detection.
- Efficiency and resilience: The integration of AI methods improves the system's efficiency and speed, making the proposed system more practical for real-world cybersecurity. Resilient cybersecurity practices are crucial to safeguard essential systems from hidden threats in backup archives and network data.

The sections are organized as follows. In Section 1, we briefly introduce the background of the project and the design at the bird's-eye-view level. In Section 2, we discuss current state-of-the-art solutions and methods and briefly compare our solution to a similar project in this area. In Section 4, we provide a detailed description of the entire system and introduce the components, methods, and models used for complex malware pattern recognition in backup systems. In Section 5, we present the results of the tests and experiments the have been carried out so far in terms of accuracy and performance. In Section 6, we discuss the results of the experiments presented in the previous section. We also propose future enhancements to the project that could potentially lead to better efficacy. In the last section, Section 7, we conclude our findings and future proposals.

2. The State of the Art

Integrated active protection against cyber threats is gaining popularity in data backup systems, protecting against data loss and downtime resulting from cyber threats such as ransomware.

Ransomware can infiltrate systems, modify files, change system configurations, and, in some cases, completely encrypt or lock the operating system, making it a critical threat on different platforms. Ransomware can be present on a computer, hiding behind legitimate-looking file names or within legitimate software packages. It often masquerades as a harmless file, such as a Portable Document Format (PDF) or Word document, to avoid detection by the user and the security software. Once executed, it can proceed to encrypt files and make other changes to the system. This deceptive tactic is part of what makes ransomware particularly dangerous and difficult to detect before it is activated. It can masquerade as system files or use file names similar to those of legitimate system files to avoid detection. This tactic can fool users and administrators into believing that files are harmless parts of the operating system.

Ransomware variants such as CryptoLocker and WannaCry [4] have been known to use deceptive practices to blend in with legitimate system processes. They may employ file names or process names that appear to be part of the operating system to avoid detection by users and security software. The exact file names they use can vary and may not be the same for different infections, as malware authors frequently change and adapt their tactics to avoid detection.

Before the actual encryption process begins, ransomware may reside on a computer using inconspicuous file names to blend in with legitimate files and avoid detection. The specific names used can be varied and are often designed to mimic the names of legitimate system files or popular software to avoid raising suspicion. Some examples could be files with names like “svchost.exe” (a legitimate Windows system process), “setup.exe” (commonly used for software installation), and “readme.txt” (a document often included with software downloads). The malicious files may also have random or generic names and be placed in common system directories such as “%SystemRoot%”, “%AppData%”, “%Temp%”, and “%ProgramFiles%”. The exact names and locations can change frequently as ransomware developers adapt to evade antivirus software and other security measures. Therefore, it is crucial to maintain updated security software that can detect such threats based on behavior, not just file names.

Several modern backup solutions now incorporate active protection features, as exemplified in Brewer’s work on ransomware [5]. These features include behavior-based detection, which analyzes running applications and processes for suspicious or malicious activity and is capable of blocking it and alerting administrators in response; anti-ransomware protection that actively monitors and can block or quarantine files or processes attempting to encrypt data or communicate with ransomware command and control servers; machine learning-based detection that employs machine learning algorithms to identify patterns and anomalies in data access and usage, facilitating the detection and prevention of cyber threats; encryption and access control measures, which encompass data encryption to safeguard against unauthorized access and access controls to limit modifications or deletions of backup data; and real-time monitoring and alerts, which provide continuous surveillance of backup activity and immediate alerts in the event of suspicious or anomalous behavior.

Numerous vendors provide data backup systems with integrated active protection against cyber threats. Table 1 provides a concise overview of the key features of selected backup solutions. Acronis Cyber Backup [6] offers a comprehensive solution that includes anti-ransomware protection, AI-driven behavioral detection, and secure data encryption. However, this solution does not take into account network communication. Furthermore, it is based on the behavior of the ransomware process, making it susceptible to latent ransomware. Carbonite [7] provides data backup and recovery solutions that feature anti-ransomware protection, machine learning-based detection, and secure data storage. Veeam [8] provides backup and recovery solutions equipped with advanced security

components such as multifactor authentication, role-based access control, and built-in ransomware protection. Druva [9] offers cloud-based data backup and recovery solutions that incorporate machine learning-based detection, data encryption, and access controls. Commvault [10] offers backup and recovery solutions with built-in security features, including malware scanning, encryption, and access controls. Veritas [11] provides data backup and recovery solutions that include ransomware protection, behavior-based detection, and secure data storage. Rubrik [12] delivers cloud-based data backup and recovery solutions featuring machine learning-based detection, anti-ransomware protection, and secure data encryption.

Table 1. Comparison of backup solutions with active protection against cyber threats.

Vendor	Key Features	Unique Aspects
Acronis [6]	Anti-ransomware protection, AI-driven behavioral detection, and secure data encryption	Comprehensive solution covering a wide range of security features
Carbonite [7]	Anti-ransomware protection, machine learning-based detection, and secure data storage	Emphasis on machine learning for threat detection
Veeam [8]	Multifactor authentication, role-based access control, and built-in ransomware protection	Strong focus on access control and authentication
Druva [9]	Machine learning-based detection, data encryption, and access controls	Cloud-based solution with a focus on machine learning
Commvault [10]	Malware scanning, encryption, and access controls	Built-in malware scanning as a primary security feature
Veritas [11]	Ransomware protection, behavior-based detection, and secure data storage	Comprehensive ransomware protection and detection capabilities
Rubrik [12]	Machine learning-based detection, anti-ransomware protection, and secure data encryption	Cloud-based solution with advanced machine learning features

3. Related Work

The main difference between our approach and the solutions from the major players in the market presented above lies in our system's wide range of detection methods that utilize both a signature-based approach and anomaly detection approach and leverage machine-learning techniques, which makes it better suited for latent malware detection. Many of the showcased solutions overlook network traffic inspection, focusing solely on process behavior features. This renders our multistage solution more robust and capable of detecting ransomware even before it exhibits any malicious actions.

The use of a combined approach involving the computation of the hash function of files and AI techniques shows significant promise in the field of file verification [13]. This method draws on a variety of techniques and combinations to effectively detect ransomware [14]. A comprehensive insight into file analysis in the context of ransomware can be found in [15].

In essence, there are two primary approaches to employing hashes. The first method involves computing the hash function of existing files, recalculating them during data backups, and comparing the original and updated values. The second approach involves the use of online databases of hash files, which can be sourced from entities such as the Computer Incident Response Center Luxembourg [16] (CIRCL) or National Institute of Standards and Technology [17] (NIST). The CIRCL, for instance, offers an Application Programming Interface (API) to request hashes in the Secure Hash Algorithm (SHA)-

1 format, which strikes a sustainable balance between performance and stability when compared to other hash functions.

Various AI algorithms find applications in ransomware analysis [18]. Machine learning (ML) algorithms can be trained on extensive datasets of ransomware samples to automatically classify new instances as malicious or benign, facilitating the swift identification of new ransomware variants by security analysts. Deep learning (DL) algorithms, including neural networks, can be employed to scrutinize ransomware behavior and identify common patterns among different ransomware families, enhancing the understanding of ransomware operations and the development of more effective mitigation strategies.

There is ongoing research on malware backup protection and ransomware detection. Most of the research, however, is focused on detecting actual ransomware encryption and is not dealing with backups. From the most recent papers, the method proposed by the authors in [19] is the most relevant to our topic. This method addresses both detection and backup protection perspectives. It aims to protect the data through immediate backups and optimizes the backup solution by avoiding backing up clean images. Simulation experiments demonstrate its effectiveness, achieving over a 50% reduction in the backup list length during heavy updates and the ability to protect data from ransomware. The paper suggests future work involving diverse experiments, protection scrum configurations, and automated setup processes.

Charmilisri et al. [20] address the surge of ransomware attacks on mobile devices and leveraging machine learning approaches and algorithms for ransomware detection. The paper reviews various detection techniques, including machine learning, behavioral analysis, and Android-specific approaches. The proposed methodology involves creating a dataset and using a random forest algorithm for classification. However, the authors labor with low accuracy and emphasize the need for a larger dataset to improve the detection of ransomware apps.

The authors in [21] introduce a novel runtime solution to defend against cryptographic ransomware. The proposed solution focuses on efficiently managing data synchronization between memory and storage subsystems, preventing maliciously encrypted data from being permanently committed to the underlying storage. The approach is robust, validated against over a thousand ransomware samples, and demonstrates a minimal performance impact.

Furthermore, in [22], the random forest (RF) was used for ransomware classification, and content-based detection algorithms for ransomware detection were presented in [14]. In the domain of data analysis, comparing the similarity of data strings, such as message hashes or fingerprints [23], often requires the application of the theory of metric space and metrics. The authors in [24] proposed a method for detecting multistage attacks using machine learning to process malware events. Inspiration can also be found in slightly different solutions, for example, for Android systems [25], where the method is based on obtaining features from the Android package kit file format. Such an approach may require efficient indexing and searching over large-scale feature vectors. However, the problem nowadays is that even source files or packages can be infected and cannot be considered a trusted source of zero infection [26]. Having explored the choices mentioned above and having summarized key aspects in Table 2, we have put forward a proposal to advance our approach by combining machine learning, hash computing, and log analysis. The contributions are summarized as follows:

- We introduced a novel and comprehensive solution designed to prevent and detect malware in backups.
- We combined multiple components and techniques, including AI, for improved malware detection and prevention.
- The preliminary simulated test demonstrates the effectiveness of our proposed system.

A comparative analysis of the most recent methods proposed is shown in Table 2.

Table 2. A comparison of backup protection and ransomware detection research.

Research	Key Features	Key Differences
Min et al. [14]	Detection method based on the file access patterns	Detection is based on the malware harmful behavior, unable to detect latent malware.
Fujinoki et al. [19]	Proactive protection solution, approach based on continuous backups	Detection is based on the malware harmful behavior, unable to detect latent malware
Charmilisri et al. [20]	Detection approach for Android, approach based on ML algorithms	Different platform and datasets
Abdulrahman et al. [21]	Detection based on actively scanning file changes before they are flushed to the permanent storage.	Detection is based on the malware harmful behavior, unable to detect latent malware.
Molina et al. [22]	ML approach based on ransomware pre-attack features	Features are related only to OS API, it does not reflect other forms of ransomware behavior.
Takey et al. [24]	Detection method based on events corresponding to the MITRE ATTACK framework	Features are related only to OS API, it does not reflect other forms of ransomware behavior.
Costa et al. [25]	Multistage detection system for Android based on ML techniques	Different platform, does not use behavioral analysis

4. The Multistage Design and Solution

The objective of the entire project is to contribute to better detection of latent malware on backup systems. It was necessary to improve the currently commonly used malware detection methods, which in this case are significantly inadequate to achieve such a goal. The path we chose to take is utilizing multistage monitoring and a subsequent evaluation of the collected data. The entire system is monitored at multiple levels using individual components, and in the event of a backup request, all these data are evaluated within a single context. This approach has its advantage; namely, it is better capable of detecting latent malware that manifests itself in the system very minimally. This multistage design allows for the evaluation of data in the context of other monitored data, thus providing a better picture of the overall health of the system. The high-level design of the entire vSafe system, as we call it, is shown in Figure 1.

The entire system is conceived as being multi-platform from the beginning and, therefore, does not rely on support for a specific virtualization or backup tool. The individual components run either as processes within a virtualized system or as separate virtual or physical machines.

The system's design encompasses a set of critical functional and non-functional requirements. Functional requirements refer to the specific features and functionalities that the system or software application must possess to serve its intended purpose effectively. In contrast, non-functional requirements establish the parameters and conditions that the system or software must adhere to in order to be considered suitable and usable.

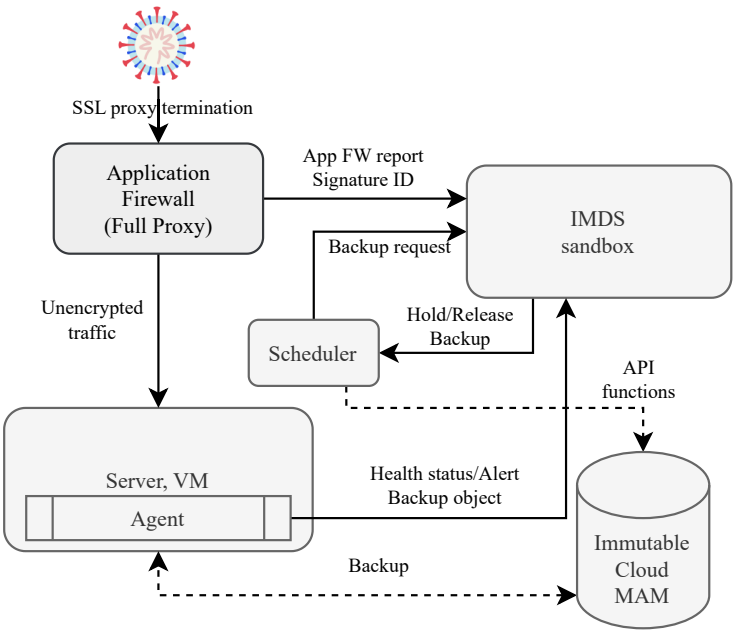

Figure 1. The general design of the vSafe system.

Within the realm of functional requirements, the system is required to inspect network traffic continuously to identify and mitigate undesirable malware and ransomware. It is also required to perform an in-depth analysis of data samples, particularly when a backup request is initiated. Additionally, the system is expected to facilitate communication with local and cloud databases by providing a user-friendly interface.

Two aspects take precedence, highlighting the non-functional requirements: the response time and processing time. The system must provide rapid responses to user queries and process data in a swift manner. Equally vital are the precision and accuracy of the results generated by the system.

Our solution is made up of three core components. This architecture offers several benefits. It can be easily integrated into any environment, providing a general design that allows for traffic monitoring and the protection of the file system. Additionally, it facilitates easy scaling, especially when working with virtual machines. Moreover, it can be easily integrated into any existing environment as it does not require anything except slightly increased resources compared to the system without our solution. The cornerstone of this solution is the Intelligent Malware Defense System (IMDS), a fusion of hardware and software. It encompasses a software agent and a full proxy device. A host in this context is typically represented by a virtual machine (VM) that operates on the Internet, serving various functions like web portals and more. Ideally, at the network edge, a proxy equipped with an application firewall is deployed. External traffic is terminated at this application firewall, and communication between the proxy and the VM remains unencrypted. This architectural choice has twofold benefits. On the one hand, it provides additional data inputs for analysis from the application firewall (proxy). On the other hand, it eliminates the need to terminate secure traffic at the VM, relieving the agent from acting as a proxy, thereby boosting the overall system performance.

The IMDS comprises several sub-functional units. The first unit is a graphical user interface, offering tools for managing other sub-functional units. Next, there is a Docker [27] sandbox implementation used to analyze individual backup images. Additionally, a database stores crucial information, including VM identification numbers from the software agent, the real-time system state of VMs, reports from the application firewall (APP FW) regarding specific VMs, and other data essential for custom analysis and logical decision making. It also interfaces with third-party APIs for database communication. The final piece of the IMDS unit is the scheduler, which is depicted separately in the figure for better clarity.

When a scheduled backup request is initiated, the IMDS sends a hold backup request and requests a backup object. Subsequently, the IMDS evaluates the information collected for the period leading up to the backup, making decisions based on the results of various analyses. If the backup is approved, the IMDS sends a release backup message to the scheduler, with all activities meticulously logged for the operator's reference. Upon receiving a safe backup response, the scheduler proceeds with the backup process according to the configured settings, utilizing the backup API for remote system operations. This controlled communication occurs between the agent and the IMDS unit, and the agent is integrated into the guest device's backup processes.

Furthermore, the system must take into account any unwanted traffic detected by the agent. The agent monitors the incoming traffic and compares it with the signature patterns obtained from an online database. If the agent issues an alert with an alert message or if the IMDS unit receives a signature ID message from the application firewall and the IMDS promptly sends a hold backup request message to the scheduler and notifies the operator interface of a potential system threat. The particular modules and analytical components of the system and their principles are described further.

4.1. vSafe File Change Detector

One of the analytical tools at our disposal is the system file change detector. Although legitimate changes to system files can occur; typically during system updates, they are relatively infrequent, and the new versions or their fingerprints are often publicly available. In the initial phase, when the system is brand new, we load hashes of the system files. Instead of recalculating the hashes directly, we retrieve them from a public database that contains hashes of commonly used system files [28,29]. When a backup request is initiated, the hash of each file is recalculated and compared to the original set. If a hash is identified that does not correspond to the original set and does not match an updated version, the backup request is denied.

To efficiently compare hashes with the original set, we have successfully utilized a similarity search model, which can identify matches within the set of original hashes. This approach is also employed for other analyzers. The details of the model are covered in Section 4.3. The advantage of using the similarity search model to find matches lies in its high efficiency compared to the classical approach, which has up to quadratic complexity. This efficiency is crucial for our application because we deal with many compared files, and practical use demands efficiency and minimal resource consumption.

4.2. vSafe Agent

Malware, particularly ransomware, does not always remain dormant after infiltrating a system; it often exhibits some activity. Although it may not immediately begin data encryption, replication, or system exploration, it frequently initiates communication with a command and control (C2) center. This communication is especially critical in the case of ransomware since it involves sending encryption keys for a specific instance to C2 servers (if they exist and the attacker intends to sell them). Shortly after infecting the system, such communication can be relatively minimal and may include basic information about the compromised system.

Suspicious communication of this nature can be detected at various layers. In cases of unencrypted traffic, a direct payload inspection is possible. However, malware developers are well aware of this issue, and communication with the C2 server is typically encrypted. In this project, we consider two levels of analysis. The first involves inspecting the JA3 fingerprint [30], a set of attributes exchanged during the establishment of Transport Layer Security (TLS) connections between the communicating parties. These attributes are readable in plain text during communication, and certain malware families have been shown to be identifiable based on the JA3 fingerprint. To achieve this, we use a publicly available database of JA3 malware fingerprints [31]. Similarly to the previous scenario, Faiss was used to identify the matches.

The second option involves monitoring traffic anomalies. The use case for our detection system is quite specific, as it is deployed in a server environment where outbound traffic from the environment should be relatively infrequent and well defined. This characteristic is advantageous when monitoring traffic and any deviations. Like the previous two analyzers, Faiss was employed to identify the matches. The model included Internet Protocol (IP) addresses, or more precisely, their hashes, for expected traffic. Then, all outbound traffic initiated by the system was analyzed for matches with known traffic. The absence of a match indicates a high probability of malicious communication.

The component of the vSafe agent that involves the techniques discussed operates directly on the virtual machine, monitoring incoming and outgoing traffic (Figure 2). The network traffic is buffered and pre-processed by a parser. The parser strips the headers and footers and extracts the data frame.

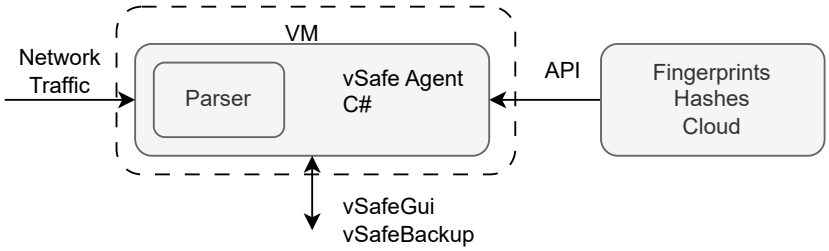

Figure 2. vSafe agent functional diagram.

4.3. vSafe Decision Model

The vSafe decision model is an integral study component and a sophisticated machine learning tool developed from the Faiss library. This model autonomously operates within its virtual machine and is essential for performing similarity searches and clustering operations, which are essential to handle complex and dense vectors. A simplified functional diagram is shown in Figure 3. It has two operational APIs. One API is used for guest-specific hash check requests. This request is sent by the central vSafeBackup component. The other API is used to retrieve hashes from both cloud and local sources.

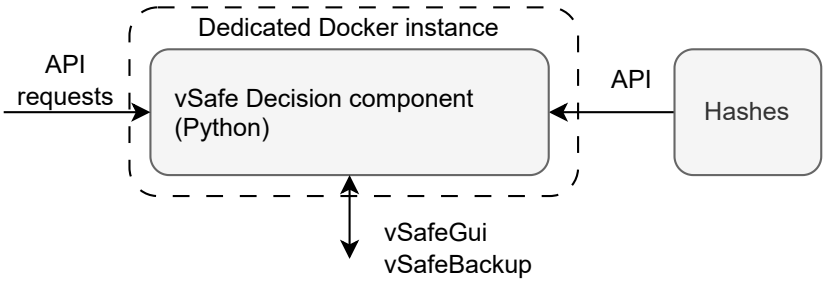

Figure 3. vSafe decision component functional diagram.

Using the Bidirectional Transformer Encoder Representations (BERT) model [32], the vSafe decision model adeptly converts hashes into dense vectors and integrates them into the Faiss index. It evaluates the similarity of an input hash to those in the index using the Euclidean distance metric, ensuring the thorough recognition of similar hashes. To improve the efficiency of data representation, the model uses the Product Quantization (PQ) algorithm for quantization, examining complex datasets that are more efficient and less resource intensive. Impressively, the vSafe decision model boasts a remarkable 99% accuracy rate in hash comparisons, due to its exceptional reliability and robustness.

4.4. vSafe Application Firewall

Another component that is not developed directly within the vSafe project but is integrated into it and included in the complete evaluation of the presence of malware is the

firewall. As mentioned in the Introduction, the TLS traffic is terminated at this firewall, and there is an application firewall (APPFW) configured at TCP/IP layer 7 for each guest [33]. This APPFW configuration is performed automatically by the Ansible tool [34] when a request is made to the vSafeGUI component to provide guest control. The APPFW inspects traffic based on known signatures. This component complements the previous vSafe agent component, which is capable of detecting deviations from standard traffic and inspecting encrypted traffic. The logs are sent to a central MDS unit, which subsequently uses them to assess in the presence of malware.

The vSafe user interface represents a GUI (graphical user interface) paradigm for administrative operations conducted through a web browser, complemented by a RESTful (Representational State Transfer) API to facilitate remote management via HTTP (Hypertext Transfer Protocol) requests. The back-end infrastructure of the current iteration is predicated on Python 3.11, while the RESTful services use the same. The client-side implementation is encapsulated within a React application, meticulously crafted in TypeScript. For persistence, a MariaDB relational database management system is utilized.

The design of the interface is in accordance with the principles of minimalism and intuitiveness. Post-authentication, users are presented with a navigable lateral menu, furnishing the capability to initiate and scrutinize the outcomes of backup operations.

Transactional requests are not processed instantaneously, reflecting the system's operational design, which accounts for the multiplicity of actions and their inherent latency. The contemplated architecture envisages a queued methodology for the management of requests. Upon the inception of a request, such as the generation of a backup, via either the Web or the REST API, the system acknowledges this initiation and sequences it within a buffer. A confirmatory HTTP status code of 200 OK is then issued. Subsequent to this acknowledgment, the request remains in a pending state within the queue, with provisions for status inquiries through the RESTful interface.

Queued requests are executed sequentially by designated worker processes. Execution results in an updated state of the system, which is subsequently communicated back to the requester via the REST API. Presently, this component is in a conceptual phase, with a focus on selecting the optimal technological framework for the queue's operational mechanics.

5. Tests and Results

The tests conducted on the various components of the system focus primarily on the vSafe detection model and the vSafe agent. The results garnered from these tests are crucial in validating the performance and accuracy of our system, which are fundamental to meeting the non-functional requirements previously outlined. Through systematic testing, we aim to ensure that the system is not only effective in its intended purpose but also efficient in its operation, minimizing the impact on the client's infrastructure.

5.1. vSafe Detection Model—Accuracy

During the preliminary assessment, variations in search accuracy were observed, contingent upon the utilization of quantization. Specifically, the Faiss model demonstrated an accuracy of 100% without the incorporation of quantization, whereas the integration of quantization led to a marginally reduced accuracy of 99%. This discrepancy in accuracy was attributed to the inherent lossiness characteristic of the quantization process, which resulted in quantized vectors that deviated from their original counterparts. Despite the discernible deviation, the disparity in accuracy was deemed inconsequential, prompting the adoption of the quantized model for subsequent experiments. These experiments encompassed diverse datasets, revealing consistent results across the entirety of the datasets.

5.2. vSafe Detection Model—Performance

The performance testing of the vSafe detection model was performed on several datasets composed of randomly generated hashes. Hashes were generated using the SHA256 hashing algorithm and split into groups of 1000; 10,000; 100,000; and 1,000,000.

The hashes were generated from the Bidirectional Encoder Representations of Transformers (BERT) neural network transmuted into vectors of 384 dimensions. The vectors were then stored in the Faiss index. The trained index was then engaged in a series of search tests conducted using the original datasets.

The search time for quantized and non-quantized search differs in the largest dataset by 54 s per one search. The result of the search time test is shown in Figure 4. In general, it is expected that the average index size will be around ten thousand hashes. In this case, the difference in search time will be around 5.5 s per search. This is unacceptable for our use case, so we will use a quantized search, even though it is less accurate.

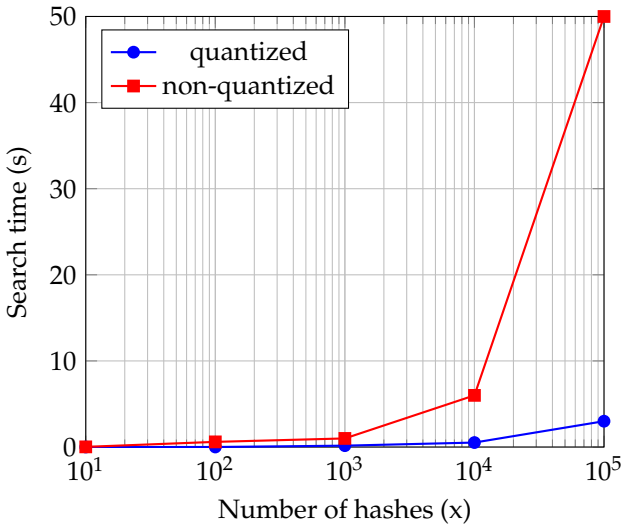

Figure 4. Search time for quantized and non-quantized search.

5.3. vSafe Agent—Performance

The dataset utilized to evaluate the performance of the TLS sniffer in the vSafe agent component was constructed through an iterative process of downloading content from the Masaryk University server. The primary objective of this data collection was to assess the time required for each download, specifically with and without the TLS sniffer enabled. In total, 1000 measurements were performed with the TLS sniffer enabled, and another 1000 measurements were performed without the TLS sniffer enabled. The purpose of these measurements was to investigate and compare the impact of the TLS sniffer on network latency. The distribution of download times is visually represented in Figure 5.

The analysis of the dataset indicates that the mean download time for the TLS sniffer-enabled sample was approximately five milliseconds higher compared to the sample without the sniffer. This difference in means suggests that the presence of the TLS sniffer may have a subtle effect on download times, albeit within a minimal margin. However, the paired *t*-test, with a hypothesized mean difference of 0, yielded a t-statistic of 1.03. When this t-statistic is compared to the critical value of 1.64 for a one-tailed test at the 0.05 significance level, it becomes evident that the results do not achieve statistical significance. In other words, there is insufficient evidence to conclude that the presence of the TLS sniffer significantly affects download times meaningfully. These findings suggest that the TLS sniffer, while introducing a slight increase in download times, does not exert a statistically significant impact on network latency.

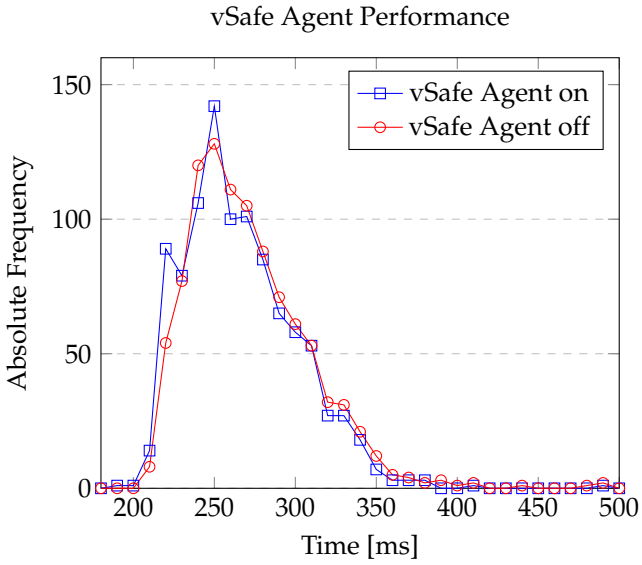

Figure 5. The vSafe agent sniffer performance.

5.4. Real Test Scenario and Results

Testing on real use cases has yet to be performed. For such testing, an isolated test polygon is created. Preliminary simulated tests have focused on using a combination of information about file system changes or the occurrence of new entries in the file system and the subsequent identification of malware fingerprints by the agent component. The results are shown in Table 3. However, to exploit the full potential, it is necessary to create so-called malware templates that contain their fingerprints and their usual behavior combined with information about the occurrence of the sample traffic from the application firewall. Such patterns will then be relearned by the model.

The table presents the outcomes of a series of tests conducted to evaluate the effectiveness of a multistage malware detection system in identifying various types of ransomware. The system uses advanced machine learning techniques and similarity search algorithms to analyze changes in system files, monitor network traffic for potential threats, and, in parallel, observe known patterns. Each entry specifies whether the ransomware was successfully detected.

Table 3. Preliminary test results of ransomware detection.

Ransomware Type	Detected (Yes/No)
WannaCry	Yes
Petya	No
NotPetya	No
Bad Rabbit	Yes
Ryuk	Yes
Sodinokibi	No
GandCrab	Yes

Petya's malware payload targets and corrupts the computer's Master Boot Record (MBR), effectively replacing the Windows bootloader. This is in contrast to Bad Rabbit malware, which masquerades as an Adobe Flash installer and is recognized by alterations in system files. Generally, the detection system in place has successfully identified ransomware when there are observable changes to the file system or new files emerge, coupled with distinctive network traffic patterns. However, in scenarios where these indicators are absent, the ransomware goes undetected.

6. Discussion

This research addresses the challenge of latent malware in backup systems, where traditional detection often fails. A novel multistage detection system is proposed, leveraging the Faiss model for enhanced file hash comparison and network traffic signature analysis. The multifaceted approach of the system combines similarity analysis with machine learning to improve malware detection, demonstrating the potential of advanced search techniques. The tests confirm the accuracy and efficiency of the system, ensuring a minimal impact on the client infrastructure. The discussion underscores the balance between detection accuracy and system efficiency, highlighting the importance of employing efficient similarity search models for practical cybersecurity applications.

One of the primary non-functional requirements for the entire system is detection accuracy and efficiency. A high number of false negative detections is problematic for obvious reasons. However, the opposite scenario is also troublesome. If there were a significant number of false positive detections that would result in rejecting uninfected backups, the effectiveness of the backup system as a whole would be significantly reduced and the practical usability of our system would be very limited. Alongside this requirement, efficiency is crucial. The entire system should impose a minimal load on the client's infrastructure, and, ideally, the client should not need to allocate additional resources. Balancing these two requirements requires a lot of work. Given the extensive need to compare binary strings from various sources, such as sandbox analysis for file hash checks or the vSafe agent component for JA3 hashes and IP addresses, a similarity search model like Faiss was tested. Faiss is a model for efficiently searching for matches in a pre-prepared database.

The vSafe detection model was rigorously tested to assess its accuracy and performance. Our preliminary tests revealed interesting dynamics concerning accuracy relative to the model's quantization state. Subsequent performance tests assessed the model's operational efficiency, handling varying-sized datasets to simulate real-world scenarios.

The performance metrics for the vSafe agent were collected through a methodical data collection process, with the objective of measuring the impact of the TLS sniffer on network latency. The statistical analysis of these results provided insight into the impact of the sniffer, helping us to understand its implications for practical deployment.

It is essential to adopt additional and diverse detection techniques to strengthen the detection capabilities against a range of ransomware strains, particularly those that do not manifest through evident changes to the file system or distinctive network traffic patterns.

7. Conclusions and Future Directions

In conclusion, this article has addressed the pressing issue of detecting latent malware in backup systems, a challenge that traditional detection methods often fail to overcome. The proposed solution offers a multistage approach, combining various components for in-depth system data analysis and network traffic. Using advanced techniques such as the Faiss model for similarity searching, the system improves the accuracy and efficiency of malware detection. The project's non-functional requirements prioritize accuracy and efficiency. Achieving a balance between minimizing false positives and false negatives is essential to ensure the practical usability of the system while imposing a minimal load on the client's infrastructure. The project has made significant progress in testing and evaluating the accuracy and performance of the system. Future steps will focus on enhancing the system and addressing any potential limitations to further improve its efficacy.

This research contributes to the field of cybersecurity by offering a comprehensive solution to detect latent malware in backups and secure critical systems, even in cases where traditional methods do not. With data and systems playing a crucial role in the digital age, safeguarding against latent malware threats is of paramount importance. Integrating advanced techniques provides a promising path to improve cybersecurity practices and protect essential systems from hidden malware threats.

In the ongoing development of the proposed malware detection system for backup systems, several promising future directions and enhancements can further strengthen its capabilities. To improve accuracy, continuous refinement of machine learning models and incorporating larger and more diverse datasets are essential. Advanced behavioral analysis techniques should be integrated to identify unusual patterns and behaviors indicative of latent malware. The real-time threat intelligence feeds and databases should be used to detect and block emerging threats proactively. Additionally, the system should evolve to detect encrypted communication more effectively, analyze encrypted traffic for suspicious patterns, and provide user training to improve cybersecurity practices. Integrating cloud-based backup solutions, compliance features, and comprehensive reporting can enhance the versatility of the system.

Furthermore, developing threat-hunting capabilities and collaboration within the cybersecurity community will improve the detection and response. Ensuring scalability, optimizing the performance, and implementing AI-driven responses are critical for the system's efficiency. Lastly, an enhanced user interface will improve the usability and user experience, making the system accessible to a broader audience. These future directions will further solidify the system's role in safeguarding critical data and systems against latent malware threats.

Author Contributions: Conceptualization, V.O. and P.N.; methodology, P.N.; software, P.N., M.H. and P.K.; validation, T.H.; investigation, T.H.; resources, V.O.; data curation, P.N. and P.K.; writing—original draft preparation, V.O. and P.N.; writing—review and editing, V.O. and P.N.; visualization, P.N.; supervision, V.O.; project administration, V.O.; funding acquisition, V.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of the Interior of the Czech Republic, Open challenges in security research, VK01030030, Data backup and storage system with integrated active protection against cyber threats.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No datasets were analyzed in this study.

Acknowledgments: This paper acknowledges the support provided by Grammarly, Writefull, and DeepL in helping to correct and improve the English of the article. The use of their tools has been invaluable in identifying and correcting grammatical errors, as well as in offering suggestions for more appropriate vocabulary and sentence structures. Thanks to their assistance, this paper has improved in clarity, concision, and overall effectiveness in communicating its ideas.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
API	Application Programmable Interface
BERT	Bidirectional Encoder Representations from Transformers
CIRCL	Computer Incident Response Center Luxembourg
C2	Command and Control
DL	Deep Learning
HTTP	Hypertext Transfer Protocol
IMDS	Intelligent Malware Defense System
IP	Internet Protocol
ML	Machine Learning
NIST	National Institute of Standards and Technology

PDF	Portable Document Format
PQ	Product Quantization
REST	Representational State Transfer
RF	Random Forest
SHA	Secure Hash Algorithm
TLS	Transport Layer Security
VM	Virtual Machine

References

1. Razaulla, S.; Fachkha, C.; Markarian, C.; Gawanmeh, A.; Mansoor, W.; Fung, B.C.M.; Assi, C. The Age of Ransomware: A Survey on the Evolution, Taxonomy, and Research Directions. *IEEE Access* **2023**, *11*, 40698–40723. [CrossRef]
2. Oujezsky, V.; Novak, P.; Horvath, T.; Holik, M.; Jurcik, M. Data Backup System with Integrated Active Protection Against Ransomware. In Proceedings of the 2023 46th International Conference on Telecommunications and Signal Processing (TSP), Prague, Czech Republic, 12–14 July 2023; pp. 65–69. [CrossRef]
3. Hervé Jegou, M.D. Faiss: A Library for Efficient Similarity Search. Available online: https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/ (accessed on 30 October 2023).
4. Connolly, L.Y.; Wall, D.S.; Lang, M.; Oddson, B. An empirical study of ransomware attacks on organizations: An assessment of severity and salient factors affecting vulnerability. *J. Cybersecur.* **2020**, *6*, tyaa023. [CrossRef]
5. Brewer, R. Ransomware attacks: Detection, prevention and cure. *Netw. Secur.* **2016**, *2016*, 5–9. [CrossRef]
6. Acronis Cyber Backup 12.5. Available online: https://www.acronis.com/en-us/support/documentation/AcronisCyberBackup_12.5/ (accessed on 5 November 2023).
7. Cloud Backup Solutions for Home and Business—Carbonite. Available online: https://www.carbonite.com/ (accessed on 31 March 2023).
8. Hoff, C. Creating Secure Backup for Ransomware Defense. Available online: https://www.veeam.com/blog/secure-backup-ransomware-defense.html (accessed on 5 November 2023).
9. Ransomware Protection and Recovery with Druva. Available online: https://content.druva.com/c/sb-ransomware-protection-recovery?x=8S3ZxU#page=1 (accessed on 30 October 2023).
10. Commvault's Immutable Infrastructure Architecture. Available online: https://cloud.kapostcontent.net/pub/6ca15136-2ef2-480d-a0b3-40880bd364f8/commvaults-immutable-infrastructure-architecture (accessed on 30 October 2023).
11. The Veritas Ransomware Resiliency Strategy—A Holistic Approach for Enterprise-Grade Storage, Data Protection, and Application Availability. Available online: https://www.veritas.com/content/dam/www/en_us/documents/white-papers/WP_ransomware_resiliency_strategy_V1551.pdf (accessed on 30 October 2023).
12. Rubrik for Ransomware Remediation Faster Ransomware Recovery from Backups That Cannot be Compromised. Available online: https://www.rubrik.com/content/dam/rubrik/en/resources/data-sheet/rubrik-ransomware-remediation.pdf (accessed on 1 November 2023).
13. Acharya, J.; Chaudhary, A.; Chhabria, A.; Jangale, S. Detecting Malware, Malicious URLs and Virus Using Machine Learning and Signature Matching. In Proceedings of the 2021 2nd International Conference for Emerging Technology (INCET), Belagavi, India, 21–23 May 2021; pp. 1–5. [CrossRef]
14. Min, D.; Ko, Y.; Walker, R.; Lee, J.; Kim, Y. A Content-Based Ransomware Detection and Backup Solid-State Drive for Ransomware Defense. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 2038–2051. [CrossRef]
15. Alzahrani, S.; Xiao, Y.; Sun, W. An Analysis of Conti Ransomware Leaked Source Codes. *IEEE Access* **2022**, *10*, 100178–100193. [CrossRef]
16. CIRCL. Available online: https://www.circl.lu/ (accessed on 28 March 2023).
17. NIST.gov—Computer Security Division—Computer Security Resource Center. Available online: https://csrc.nist.gov/ (accessed on 10 November 2023).
18. Faruk, M.J.H.; Shahriar, H.; Valero, M.; Barsha, F.L.; Sobhan, S.; Khan, M.A.; Whitman, M.; Cuzzocrea, A.; Lo, D.; Rahman, A.; et al. Malware detection and prevention using artificial intelligence techniques. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; pp. 5369–5377.
19. Fujinoki, H.; Manukonda, L. Proactive Damage Prevention from Zero-Day Ransomwares. In Proceedings of the 2023 5th International Conference on Computer Communication and the Internet (ICCCI), Fujisawa, Japan, 23–25 June 2023; pp. 133–141. [CrossRef]
20. Charmilisri, A.; Harshi, I.; Madhushalini, V.; Raja, L. A Novel Ransomware Virus Detection Technique using Machine and Deep Learning Methods. In Proceedings of the 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 17–19 May 2023; pp. 8–14. [CrossRef]
21. Elkhail, A.A.; Lachtar, N.; Ibdah, D.; Aslam, R.; Khan, H.; Bacha, A.; Malik, H. Seamlessly Safeguarding Data Against Ransomware Attacks. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 1–16. [CrossRef]
22. Molina, R.M.A.; Torabi, S.; Sarieddine, K.; Bou-Harb, E.; Bouguila, N.; Assi, C. On Ransomware Family Attribution Using Pre-Attack Paranoia Activities. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 19–36. [CrossRef]

23. Novak, P.; Oujezsky, V. Detection of Malicious Network Traffic Behavior Using JA3 Fingerprints. In *Proceedings II of the 28th Conference STUDENT EEICT 2022*; Novák, A.P.V., Ed.; Brno University of Technology, Faculty of Electrical Engineering and Communication: Brno, Czech Republic, 2022; pp. 194–197.
24. Takey, Y.S.; Tatikayala, S.G.; Patil, M.U.; R, L.E.P.; Samavedam, S.S. Real Time Multistage Attack Detection Leveraging Machine Learning and MITRE Framework. In Proceedings of the 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 16–17 December 2022; pp. 1226–1230. [CrossRef]
25. Costa, L.d.; Moia, V. A Lightweight and Multi-Stage Approach for Android Malware Detection Using Non-Invasive Machine Learning Techniques. *IEEE Access* **2023**, *11*, 73127–73144. [CrossRef]
26. Jibilian, I.; Canales, K. The US Is Readying Sanctions against Russia over the SolarWinds Cyber Attack. 2020. Available online: https://www.businessinsider.com/solarwinds-hack-explained-government-agencies-cyber-security-2020-12?op=1 (accessed on 26 January 2024).
27. Docker: Accelerated, Containerized Application Development. Available online: https://www.docker.com/ (accessed on 31 March 2023).
28. National Institute of Standards and Technology. National Software Reference Library (NSRL). 2016. Available online: https://www.nist.gov/itl/ssd/software-quality-group/national-software-reference-library-nsrl (accessed on 6 November 2023).
29. OPSWAT. Metadefender Cloud API v4 Documentation. Available online: https://docs.opswat.com/mdcloud/metadefender-cloud-api-v4 (accessed on 6 November 2023).
30. Althouse, J. TLS Fingerprinting with JA3 and JA3S. Available online: https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967 (accessed on 28 October 2023).
31. Ja3 fingerprints Database. Available online: https://sslbl.abuse.ch/ja3-fingerprints/ (accessed on 30 October 2023).
32. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
33. F5. What Is a Web Application Firewall (WAF)? Available online: https://www.f5.com/glossary/web-application-firewall-waf, (accessed on 6 November 2023).
34. Red Hat, Inc. Ansible Runner Documentation. Available online: https://ansible-runner.readthedocs.io/en/stable/index.html, (accessed on 6 November 2023).