electronics

MDPI

Article

Evidence Preservation in Digital Forensics: An Approach Using Blockchain and LSTM-Based Steganography

Mohammad AlKhanafseh [1] and Ola Surakhi [2,*]

[1] Department of Computer Science, Birzeit University , Birzeit P.O. Box 14 , West Bank, Palestine; malkhanafseh@birzeit.edu
[2] Cybersecurity Department, American University of Madaba, Madaba 11821, Jordan
* Correspondence: o.surakhi@aum.edu.jo

Abstract: As digital crime continues to rise, the preservation of digital evidence has become a critical phase in digital forensic investigations. This phase focuses on securing and maintaining the integrity of evidence for legal proceedings. Existing solutions for evidence preservation, such as centralized storage systems and cloud frameworks, present challenges related to security and collaboration. In this paper, we propose a novel framework that addresses these challenges in the preservation phase of forensics. Our framework employs a combination of advanced technologies, including the following: (1) Segmenting evidence into smaller components for improved security and manageability, (2) Utilizing steganography for covert evidence preservation, and (3) Implementing blockchain to ensure the integrity and immutability of evidence. Additionally, we incorporate Long Short-Term Memory (LSTM) networks to enhance steganography in the evidence preservation process. This approach aims to provide a secure, scalable, and reliable solution for preserving digital evidence, contributing to the effectiveness of digital forensic investigations. An experiment using linguistic steganography showed that the LSTM autoencoder effectively generates coherent text from bit streams, with low perplexity and high accuracy. Our solution outperforms existing methods across multiple datasets, providing a secure and scalable approach for digital evidence preservation.

Keywords: blockchain; evidence preservation; forensics; long-short term memory; steganography

Citation: AlKhanafseh, M.; Surakhi, O. Evidence Preservation in Digital Forensics: An Approach Using Blockchain and LSTM-Based Steganography. *Electronics* **2024**, *13*, 3729. https://doi.org/10.3390/electronics13183729

Academic Editor: Aryya Gangopadhyay

Received: 9 August 2024
Revised: 31 August 2024
Accepted: 6 September 2024
Published: 20 September 2024

1. Introduction and Basic Concepts

With an increasing reliance on digital devices, the performance of smart technology like smartphones and tablets continues to advance. This progress in smart devices has sparked significant demand among users of varying technical expertise. These devices are used both by individuals well-versed in technology and by those with limited or no background in this realm. Consequently, the incidence of crimes committed using these devices has risen considerably, necessitating thorough investigation processes, making the role of digital forensics increasingly pivotal.

Digital forensics involves investigating digital crimes, focusing on recovering, preserving, and analyzing electronic evidence crucial for supporting criminal inquiries. In this context, preserving evidence stands as a cornerstone of the investigative process. Given the ongoing surge in digital data production, ensuring the integrity, authenticity, and admissibility of digital evidence is more crucial than ever before, demanding robust solutions to prevent tampering.

Preserving evidence presents various challenges beyond mere retention; it entails ensuring the credibility of evidence throughout the investigation's lifecycle, from seizure to its presentation in court. This might include the need to have a multi-layered security level. These layers act as a series of hurdles, making it significantly harder for unauthorized individuals to access, tamper with, or corrupt critical evidence. Secure frameworks with multi-level security enable the controlled sharing of evidence between authorized

personnel and long-term preservation of evidence in a tamper-proof manner, which is crucial for legal proceedings that may extend over months or even years. Blockchain technology offers a decentralized and immutable ledger solution, ensuring highly secure and transparent records, timestamps, and preservation of digital evidence and related media in forensic investigations [1]. Blockchain can play a crucial role in securing digital evidence and enhancing the solutions. The potential benefits for securing evidence by using blockchain technology are as follows: (1) Enhanced security for evidence storage due to its decentralized nature, immutability, and transparency, which prevent unauthorized tampering and ensure data integrity. Each piece of evidence would be cryptographically hashed and stored on a distributed network. (2) Creates an unbreakable chain of custody, ensuring the court can be confident that the evidence presented is authentic and has not been tampered with throughout the investigation. (3) The immutable ledger of a blockchain allows for easy tracing of evidence movement. Integrating blockchain technology can elevate the forensic process by enhancing evidence collection, preservation, and analysis in digital forensics, ultimately refining the entire investigative procedure.

On the other hand, transferring large amounts of digital evidence across networks can be a challenge, especially when considering security concerns. Steganography can indeed be a helpful tool in this specific scenario. Steganography hides the evidence within another seemingly innocuous file, like an image or text. This can be beneficial if there are concerns about someone intercepting the data during transfer. By disguising the evidence, it becomes less likely to attract attention. The use of steganography to conceal evidence within a carrier file adds an extra layer of covertness during transfer, resulting in a more robust solution to ensure the evidence remains unaltered.

This paper endeavors to introduce a novel framework for preserving evidence in digital forensics. The proposed solution aims to bolster the security, privacy, and authenticity of digital evidence throughout the investigative process. It leverages various advanced solutions in cybersecurity, including steganography, blockchain, and cryptography, to fortify the investigation's integrity and efficacy.

In summary, the integration of blockchain and LSTM-based steganography provides a balanced approach to enhancing both the integrity and privacy of digital evidence, addressing some of the core challenges in digital forensics. It is motivated by the distinct advantages each technique offers in addressing critical challenges in digital forensics:

1. Blockchain for Integrity and Immutability: Blockchain technology is renowned for its ability to create an immutable ledger, ensuring that once evidence is recorded, it cannot be tampered with. This feature is crucial in digital forensics, where maintaining the integrity of evidence is paramount. By incorporating blockchain, the framework provides a transparent and secure chain of custody, ensuring that every action related to the evidence (e.g., access, modifications, transfers) is logged in a verifiable, timestamped manner. This greatly reduces the risk of data alteration or manipulation, a key concern in forensic investigations.
2. LSTM-based Steganography for Privacy and Confidentiality: The use of Long Short-Term Memory (LSTM)-based steganography in this framework is designed to enhance the privacy of digital evidence. Steganography allows for the covert embedding of information within other forms of media, making the existence of the evidence itself less detectable. The LSTM model is particularly effective at encoding data in ways that are difficult to detect or reverse-engineer, thus providing a high level of confidentiality. This is critical during the transmission or storage phases, where unauthorized access could compromise the investigation.
3. Complementary Synergy: Blockchain and steganography are complementary techniques, where blockchain secures the integrity and traceability of the evidence, and steganography protects its confidentiality. Together, they provide a comprehensive solution to ensure the evidence is not only immutable but also shielded from unauthorized access or exposure. The framework addresses both of these essential

aspects, strengthening the overall trustworthiness and efficacy of digital evidence in forensic processes.

In this study, we tested the model using the tiny_shakespeare dataset, which was preprocessed through tokenization. This process involves breaking down long text strings into smaller units, first by dividing text into sentences and then into words. Redundant words were eliminated to create a refined vocabulary used in the bit block mapping process. Additionally, we incorporated another dataset, Email Communication Logs, to assess the model's performance in forensic evidence scenarios. This dataset provides a different context to evaluate the model's effectiveness in handling real-world data. Our findings indicate that the LSTM autoencoder performs exceptionally well in generating natural-sounding text from a secret bit stream, achieving low perplexity and strong coherence metrics. This effectiveness demonstrates the model's suitability for text-hiding purposes, ensuring that the embedded message is secure and remains indistinguishable from ordinary text. Furthermore, the model exhibits high precision in recovering the original bit stream, emphasizing its reliability in maintaining the integrity of concealed information.

1.1. Concepts of Blockchain Technology

Satoshi Nakamoto [1] introduced blockchain technology in a white paper in 2008 for the first time. Blockchain represents a decentralized peer-to-peer (P2P) network with a distributed ledger that is both immutable and transparent [2]. In this blockchain system, the distributed ledger takes the form of interconnected blocks, where each block is connected to the previous one through its cryptographic hash [3]. Each block is assigned a unique block number and a timestamp. Within each block, multiple transactions are structured in a Merkle tree format, and each transaction is secured through cryptographic signatures using an asymmetric digital key. The rules for managing and maintaining the distributed ledger are determined by a governance mechanism known as the consensus algorithm [4].

Blockchain can be either public or private. Public blockchains are accessible to everyone, allowing anyone to view the ledger. In contrast, private blockchains restrict ledger access to specific members. Permissionless blockchains enable anyone to engage in transactions and join the consensus process, whereas permission-based blockchains restrict these activities to chosen members [5]. To efficiently and securely encode blockchain data, a Merkle tree is employed. Additionally, consensus algorithms play a crucial role in achieving synchronization and establishing agreement among the numerous nodes within a blockchain network [5].

A Merkle tree aggregates all the transactions within a block and creates a digital fingerprint representing the entire set of operations. This fingerprint enables users to verify the presence or absence of specific transactions within the block. The Merkle tree is structured as a binary tree, where each leaf node represents the hash of an individual transaction, and each non-leaf node represents the hash of a combination of its child nodes. Moreover, the Merkle tree's root serves as a means to verify the data within it. The Merkle root employs a straightforward mathematical process to validate the data contained in the Merkle tree [6].

The consensus algorithm plays a critical role in establishing the rules for maintaining the distributed ledger [7]. The consensus algorithm is essential to achieving unanimous agreement across the entire blockchain network when determining the validity of each individual block because, in a blockchain, there is no central authority responsible for validating transactions or blocks. The literature showcases a variety of consensus algorithms, each with its unique characteristics and implementations [8].

Proof-of-Work (PoW) is a consensus algorithm that is used in Bitcoin as well as Ethereum and imposes a requirement on the peers in the blockchain network to solve a challenging mathematical problem [9]. In a PoW system, blockchain validators continuously run the data from a block's header through a cryptographic hash function. With each iteration, validators incorporate a random number known as a nonce. Determining the

data added to the next block in a blockchain using PoW necessitates significant energy consumption and computational resources. The participants competing to solve the problem and validate blocks in the PoW are referred to as miners. They are rewarded based on the computational resources they expend [10].

Many consensus algorithms have been developed in the literature to overcome PoW cost issues such as Proof-of-Stake (PoS), Delegated-Proof-of-Stake, the Byzantine consensus algorithm, and more [11].

1.2. Digital Forensics

Digital forensics is the procedure of identifying, preserving, analyzing, and presenting digital evidence in a manner that meets the requirements for admissibility in a legal court, i.e., it must maintain the integrity of the collected evidence, ensuring its reliability and usability in a legal case [12]. The goal of a digital forensics investigation is to preserve the evidence as it exists while also uncovering information that helps the investigator reconstruct past events and understand not just how, but also why, they occurred the way they did.

The digital forensic process entails five steps: identification, preservation, analysis, documentation, and presentation [13], as shown in Figure 1.

Figure 1. The digital forensics process.

Identification serves as the initial phase in any digital forensic investigation. During this stage, the investigator or investigative team is tasked with recognizing the scope of the evidence contained on the device, its storage locations, and the specific formats in which it is stored. Digital evidence can encompass a wide range of formats, such as text messages, emails, images, videos, web search histories, documents, transactions, and more. This evidence can be found on various devices, including computers, smartphones, tablets, fitness trackers, and numerous other digital platforms.

After identification, the next step in digital investigations is preservation. This phase involves the isolation and secure safeguarding of data, as well as the creation of a copy or image that can be subject to analysis and investigation. Preserving the original evidence in its unaltered state is crucial in digital investigations to ensure its admissibility as evidence in a court of law.

Analysis is the phase in a digital forensic investigation during which the forensic scientist or investigator pieces together fragmented data to construct a comprehensive narrative of the events that transpired during the crime or matter under investigation. Forensic experts primarily rely on the evidence, drawing from their experience and expertise. It often requires multiple attempts and examinations to formulate a well-substantiated theory about the unfolding of the crime.

Documentation involves the creation of a record that compiles the data to be presented in court or any other forum where the investigation is being resolved. It comprises a narrative reconstruction of the events in question, intertwined with the evidence that bolsters the theory. This documentation should be persuasive to an external party tasked with determining guilt or innocence.

Presentation marks the concluding phase of the digital forensic process. The investigator utilizes the documentation to articulate the conclusions they have drawn regarding the event in question. Whether the conclusion is conveyed in a courtroom or through a written report, the investigator's task is to translate their expert findings into a clear and comprehensible narrative that can be understood and assessed by individuals who are not experts in the field, relying on the details and evidence provided.

The investigative process commences immediately upon the incident being reported or a crime being detected [14]. Subsequently, the investigator proceeds through the steps depicted in Figure 1. Initially, the investigation commences with the identification of the machine or object associated with the alleged crime or violations. Once the crime is detected, the investigator begins gathering evidence from the objects identified as being involved in the offense. Following this, the investigator scrutinizes these objects and compiles a report detailing the findings. Ultimately, the last step involves reporting these findings and apprehending the suspect [15].

Computer Forensics Domains

The computer forensics domains can be divided into eight main areas as follows:

1. Operating system forensics.
2. Disk and file system forensics.
3. Live memory forensics.
4. Web forensics.
5. Email forensics.
6. Network forensics.
7. Multimedia forensics.
8. Others.

Operating system forensics is the procedure of extracting valuable information from the operating system of a computer or mobile device under investigation [16].

The file system is of utmost importance in computing, as it ensures the organization and accessibility of data. Without it, files would become disorganized, making it impossible to determine their location, starting point, or endpoint. Each instance of a file system has a distinct size, but it can be processed by any computer supporting that particular file system type. Different file systems come with varying structures, logic, speed, flexibility, security, size, and other attributes [17]. File systems encompass key components such as file names, directories, metadata, and space management. Analyzing a file system relies on the data contained within a partition or disk. This usually entails processing data to extract the contents of a file or retrieve the contents of a deleted file [18].

Live memory, RAM (Random Access Memory), facilitates the access and processing of various types of information, handles, open files, decrypted data, registry entries, user passwords, user activities, as well as connection and session details [19]. RAM permits data access in a manner that can unveil transparent information that might be otherwise concealed. This capability is instrumental in revealing hidden processes, detecting malware attempting to conceal information, and identifying the use of various tool kits [20].

In web forensics, forensic information can be obtained from various sources like web storage records, browsing sessions, search histories, and complete user activity logs. Each operating system (OS) and web browser has its unique method for retaining these records, which can be analyzed to trace and investigate criminal activities [21].

Email forensics is the process of gathering evidence from emails. Email serves as an electronic means of communication over the Internet, facilitating the exchange of messages, files, documents, and various transaction-related elements. This process involves the examination and analysis of email content to extract valuable information for investigative purposes [22].

Network forensic analysis centers on the monitoring of network traffic and the investigation of the sources of potential attacks. This process involves the collection and analysis of data related to network activities and events, intending to identify and mitigate security breaches or unauthorized access [23,24].

In contemporary times, digital visual media has become one of the primary modes of communication. Digital images are the subject of numerous digital investigations, as some of them may contain illicit content [25]. This type of analysis seeks to uncover details about the image's origin, including its location and the individuals depicted in it. Image analysis

also involves scrutinizing images for potential evidence of steganography, which is the practice of concealing information within digital media.

There are various other domains within computer forensics. For example, cloud forensics is employed to investigate crimes committed using cloud platforms, focusing on digital evidence hosted in cloud environments. Database forensics, on the other hand, is used to examine data storage systems and address privacy-related crimes, involving the analysis of databases to uncover evidence of illicit activities.

The rising prevalence of cybercrimes worldwide, which vary in nature and complexity, including content forgery, financial data fraud, and even cyber terrorism involving large groups and government actors, has underscored the necessity for computer forensic algorithms, solutions, and tools. In response to these evolving and multifaceted threats, the development and advancement of forensic techniques and technologies have become imperative for effectively combating and investigating cybercrimes.

Blockchain technology has the potential to provide various applications for digital forensics investigations. These applications encompass evidence collection, preservation, validation, and analysis. Researchers and investigators can harness the capabilities of blockchain for digital forensics, as it allows for traceability and ensures the immutability of records. This can be particularly valuable in maintaining the integrity and reliability of digital evidence throughout the investigative process.

1.3. Steganography Technique

Steganography encompasses methods used to conceal confidential information within other digital media, essentially hiding it in plain sight. Some definitions characterize it as the practice of embedding one piece of information within another in a way that makes detection challenging, akin to a form of camouflage that remains unseen by unintended recipients or intruders [26]. Within the realm of information-hiding techniques, steganography is regarded alongside methods like watermarks and cryptography [27]. The primary objective of these techniques is to ensure secure transmission and communication across insecure channels.

However, the goals of steganography differ from those of watermarks. Watermarks primarily aim to provide proof of ownership or identification, while steganography focuses on achieving confidentiality, integrity, and privacy through encryption. Notably, research indicates that steganography is utilized as an anti-forensics technique, enabling the hiding of evidence and complicating investigation processes [28]. Essentially, steganography operates differently from cryptography, which revolves around the art of rendering secret messages incomprehensible unless the specific key to decoding them is known.

Steganography, as a field, is broadly categorized into two primary divisions: linguistic steganography and technical steganography. Linguistic steganography involves concealing text within a text cover message, while technical steganography utilizes diverse types of media as cover messages.

The most common methods employed in steganography are image steganography, text steganography, audio steganography, and video steganography. Each method uses a specific carrier or cover media. For instance, in text steganography [29], the carrier medium utilized is text. Within text steganography, the process of embedding a secret message into the carrier content can be executed through various techniques. These techniques might include altering spacing, selecting different fonts, or manipulating letter cases to conceal the hidden message within the text.

Indeed, the counterpart to steganography is steganalysis, which focuses on uncovering or detecting the embedded message within cover media. Steganalysis involves the investigation and detection of hidden data concealed using steganographic techniques. This field of study is generally divided into two primary categories.

Passive steganalysis aims to categorize a cover medium as either “stego” (having hidden data) or not and seeks to identify the steganographic embedding algorithm. On the other hand, active steganalysis goes a step further. In addition to determining whether

a cover medium contains hidden data and identifying the embedding algorithm, active steganalysis endeavors to estimate the length of the embedded message and, ideally, extract it from the cover medium [30].

2. Related Works

The study presented in [31] introduced an innovative framework known as the "Forensics Chain for Evidence Preservation System for IoT-based smart city security" to address the issues faced by forensic investigators, such as the risk of a single point of failure and potential evidence alterations. These issues arise when digital evidence is stored on cloud servers—it elevates the risk of evidence tampering and unauthorized sharing with malicious third parties.

Building on a different aspect, the work discussed in [32] proposes a privacy-preserving digital forensics (P2DF) framework. The framework's objective is to safeguard the digital crime scene for subsequent validation through the synchronization and analysis of evidence. Following the confiscation of suspects' digital media, a thorough bit-by-bit imaging process is carried out on the original data contents. The integrity of the secure digital images is verified at each stage using MD5 or SHA-1 hash techniques to promptly detect any alterations in the copied image.

Another noteworthy contribution is highlighted in [33], where a digital forensic framework that leverages case information, case profile data, and expert knowledge to automate the digital forensic analysis process is developed. It also employs machine learning to identify the most pertinent evidence, all while safeguarding data privacy. This approach enhances the overall efficiency of digital forensic investigations without compromising the integrity and admissibility of the evidence.

In the context of Internet-of-Things (IoT) devices, the framework presented in [34] leverages blockchain technology to secure multimedia evidence. The authors have put forth the "BEvPF-IoT" method, which is a framework for preserving multimedia evidence in the context of Internet-of-Things (IoT) devices. This method is based on blockchain technology. The framework includes the establishment of a secure and cost-effective environment through the utilization of IPFS and the Ethereum blockchain. This implementation aims to improve network transparency and accountability during the forensic examination of digital multimedia evidence.

The work in [35] introduced a blockchain-powered solution tailored for the smart home sector, addressing the tasks of gathering and safeguarding digital forensic evidence. The system employs a private forensic evidence repository to store collected evidence, alongside a permissioned blockchain framework that delivers security functionalities such as integrity, authentication, and non-repudiation.

The blockchain technology was again used by the authors in [36] to provide a solution for evidence preservation of forensics technology. The proposed framework called IoTFC can deliver a guarantee of traceability and track provenance of evidence items. Details of evidence identification, preservation, analysis, and presentation will be recorded in chains of blocks. The IoTFC can increase trust of both evidence items and examiners by providing transparency of the audit train.

Similarly, ref. [37] introduced "ForensiBlock" which is a dedicated private blockchain solution. ForensiBlock guarantees thorough and transparent documentation throughout the investigative journey, encompassing stages such as data extraction, access control, and tracking of data versions.

In a different approach, ref. [38] proposed a solution based on the theory of ontology to preserve the privacy in the area of digital forensics by abstracting the privacy attributes in digital forensics scenarios.

Lastly, ref. [39] proposed a practical and secure CustodyBlock (CB) model, which employs a private blockchain protocol and smart contracts. This model is designed to facilitate the management, transfer, analysis, and monitoring of evidence in a reliable

manner. The integration of smart contracts within CB serves to automate and enhance the evidence preservation and handling processes, making them more efficient and secure.

The reviewed literature highlights several advancements in digital forensics, each offering unique benefits and facing specific challenges.

1. Blockchain-Based Solutions: Several frameworks utilize blockchain technology to enhance the integrity and immutability of forensic evidence [40]. These solutions effectively address issues like unauthorized alterations and single points of failure by leveraging blockchain's decentralized and tamper-proof nature. For example, frameworks that incorporate blockchain with IPFS or Ethereum ensure secure and transparent storage of multimedia evidence, improving accountability. However, these systems often encounter scalability issues and high implementation costs, which can limit their practical adoption in large-scale forensic operations.
2. Privacy-Preserving Digital Forensics: Privacy-preserving frameworks focus on maintaining data integrity through methods such as bit-by-bit imaging and hash verification. These approaches are adept at ensuring the authenticity of digital evidence by detecting alterations during the imaging process. While effective in preserving data privacy, these methods may not fully address the security of evidence throughout its entire lifecycle or during complex forensic analyses, making them less suitable for dynamic forensic environments [41].
3. Automation and Machine Learning: Automation and machine learning frameworks aim to enhance efficiency and accuracy in forensic investigations. By automating evidence analysis and employing machine learning algorithms to identify relevant data, these systems can significantly speed up the investigative process. However, they may face challenges related to data privacy and the generalization of models across various types of evidence. The reliance on quality training data also impacts the performance and effectiveness of these methods [42,43].
4. Private Blockchain and Smart Contracts: Solutions utilizing private blockchains and smart contracts provide secure environments for managing and automating evidence handling. These frameworks offer benefits such as improved data management and automated processes, reducing human error. Despite these advantages, private blockchains may have limited flexibility compared to public blockchains and can involve higher implementation costs, potentially restricting their adaptability and affordability.
5. Ontology-Based Privacy Preservation: Ontology-based approaches abstract privacy attributes to safeguard data in forensic contexts. This method offers flexibility and adaptability by defining and managing privacy policies tailored to specific needs. Nonetheless, the complexity of implementing ontology-based systems and their limited integration with other technologies like blockchain or steganography can constrain their effectiveness in comprehensive forensic solutions [44].

By analyzing the previous works, it is evident that the advancement of blockchain technology, with its inherent qualities of integrity and immutability, has significantly influenced the development of privacy-preserving solutions for forensic evidence. Blockchain technology has spurred numerous innovative approaches aimed at safeguarding sensitive data and transactions, ensuring their authenticity, and preventing unauthorized alterations. This trend highlights the growing recognition of blockchain's potential to provide robust privacy preservation in forensic contexts. However, while existing frameworks effectively utilize blockchain for secure storage and management, our proposed approach introduces a novel integration of autoencoders with steganography and blockchain technology. This multi-layered framework enhances evidence security by incorporating advanced data embedding techniques with autoencoders, which offer improved data concealment and robustness against tampering. Additionally, our approach integrates steganographic methods to embed forensic evidence within cover text, followed by secure storage on a blockchain, thereby combining multiple layers of protection and innovation. This comprehensive

solution addresses both the concealment and integrity of evidence, advancing the field beyond the capabilities of current blockchain-based systems.

3. The Proposed Model

The proposed model consists of four main parts:

1. Evidence Preprocessing.
2. Evidence Fragmentation.
3. Cover File Creation and Evidence Hiding.
4. Evidence Storing.

Each phase is designed to ensure the integrity, privacy, and security of digital forensic evidence throughout its lifecycle. The following principles guided the development of the model:

1. Efficient Evidence Management: To handle diverse evidence types effectively, the first step in the framework is to categorize evidence based on its data type (e.g., image, text, video, PDF). This organization not only enhances the management of evidence but also allows for more efficient and accurate processing by tailoring the subsequent steps to the nature of the data.
2. Balancing Security and Manageability: Evidence is segmented into 10 parts, a threshold chosen to enhance security through encryption and embedding, while maintaining data integrity and ease of processing.
3. Privacy via LSTM-based Steganography: LSTM-generated stegotext is used to conceal forensic evidence securely, ensuring the privacy of the data during transmission and storage.
4. Blockchain for Tamper-Proof Storage: The final steganography file is stored on a blockchain to provide immutable, verifiable records of the evidence, guaranteeing its integrity and authenticity.

In the initial phase of our proposed model, the collected evidence is assembled for the preprocessing phase. During this phase, the evidence is organized into groups according to their data types, which could be image, text, video, PDF, etc. A machine learning method is employed to cluster the evidence data.

Following this preprocessing, the next phase begins. In this phase, each piece of evidence is divided into a set of ten chunks equal to 10. This threshold was selected based on a balance between security and manageability. Dividing the evidence into ten parts ensures that each chunk is small enough to be securely encrypted and embedded, while still maintaining the integrity and coherence of the original data. This segmentation allows for more efficient processing and enhances the robustness of the steganography method by distributing the evidence across multiple segments, making unauthorized reconstruction more difficult. Then, we employed a previously published method for creating the cover file (stegotext) in the third phase of the proposed model. The authors proposed this method in [45]. They utilized Long Short-Term Memory (LSTM) to generate steganographic text for hiding information. The proposed model automatically generates stegotext using an LSTM. The secret data is then embedded in the generated stegotext.

The stegotext file will be utilized in the third phase of our model to conceal forensic evidence within it and create the steganography file, resulting in the creation of transaction data to be stored within the blockchain in the form of a steganography file.

Lastly, the generated file is stored in the blockchain database following the execution of the blockchain mining process. The key features of the proposed model can be summarized as follows:

1. Ensure a heightened level of integrity, confidentiality, and privacy for digital evidence.
2. Employing a solution like blockchain offers an immutable safeguard, ensuring that any attempt to tamper with the content of a single piece of evidence would necessitate extensive effort.

3. Utilizing steganography techniques to conceal evidence in forensics is crucial due to its capacity to obscure sensitive information within seemingly innocuous data.

The detailed steps are as follows:

Preprocessing Phase: Organize collected evidence into data-type-specific groups and use machine learning techniques for data clustering.

Fragmentation Phase: Each piece of evidence is divided into a set of chunks.

Cover File Creation and Evidence Hiding: Create the initial stegotext cover file using the LSTM method, then embed the evidence chunks into the cover file.

Final Step: Store the stegotext file within the blockchain database after completing the mining process.

Evidence Preprocessing

From the pool of evidence available, a specific piece stands out as the starting point for initiating the preservation process. After receiving the evidence, it undergoes a series of preprocessing procedures to be categorized into the appropriate dataset based on its type. The memory pool is a data structure that holds the evidence that has not yet been included in the mining process.

During this stage, each piece of evidence undergoes a series of data preprocessing procedures. The evidence can exist in various formats, including text, images, audio, video, and more. To manage these diverse data types, protocols can be implemented for regulation. In our proposed model, we handled text data type. We integrate and organize sets of evidence of the same data type into distinct groups, ensuring that text data is collected in one group while another data type is collected in another.

Evidence Fragmentation

Upon completing the data preparation process, the evidence is fragmented and divided into smaller files known as chunks.

Initially, the data is partitioned into ten distinct chunks. This threshold was selected based on a balance between security and manageability. Dividing the evidence into ten parts ensures that each chunk is small enough to be securely encrypted and embedded, while still maintaining the integrity and coherence of the original data. This segmentation allows for more efficient processing and enhances the robustness of the steganography method by distributing the evidence across multiple segments, making unauthorized reconstruction more difficult.

Each of these chunks is then converted into a bit stream based on the ASCII representation for the letters in the message, such as if the message is a letter A, the bit representation for this 1000001.

Each bit stream is then divided into small blocks. The number of blocks depends on the chosen block size. For example, if each block contains two bits, the bit stream "1000001" is divided into blocks by dividing the total number of bits in the message by the number of bits per block. As a result, the original message is segmented into blocks based on this bit distribution. In the provided bit stream, the distribution across blocks would be: "10" as the first block, "00" as the second block, "00" as the third block, and "01" as the fourth block. If the number of bits per block increases, the number of blocks for the given message will be reduced. For instance, if each block contains three bits, the bit stream "1000001" would be divided into "100", "000", and "1". The probability of combining more bits into one block is based on the equation: Number of possible blocks = $2^{b}itsperblock$.

Cover File Creation and Evidence Hiding

During this stage, the objective is to generate a cover file that will serve as a steganography container for concealing evidence within it. This is accomplished on the principles of linguistic steganography model which is proposed by [45]. The technique relies on embedding a hidden piece of information within a piece of text that appears to be natural language ("stegotext") from LSTM model. The LSTM model will generate words in natural language based on a bit-block sequence that is generated in the previous phase. LSTM networks were selected for their unique capabilities in handling sequential data and capturing long-range dependencies, which are crucial for generating realistic and coherent stegotext.

In steganography, where the goal is to embed information within text without detection, LSTMs offer distinct advantages:

1. Natural Text Generation: LSTMs are adept at creating natural-sounding sequences, which helps in maintaining the integrity and readability of the cover text. This is essential for ensuring that the embedded evidence does not disrupt the text's appearance or function.
2. Long-Term Context Preservation: The ability of LSTMs to remember and leverage long-term dependencies allows for more effective and contextually appropriate embedding of information, reducing the likelihood of detectable anomalies.
3. Proven Effectiveness: Prior studies have validated the effectiveness of LSTMs in generating high-quality text, reinforcing their suitability for our steganographic method.

The proposed algorithm in [45] started from 'vocabulary'—this set includes all possible tokens that might appear in the stegotext. Typically, these tokens are words, but they can also be punctuation marks. The process involves dividing the tokens among the blocks without any overlapping, creating disjoint token sets. This involves randomly selecting tokens from the vocabulary without replacement, ensuring each token appears exactly once in a single block. Consequently, the number of tokens in each block equals the total size of the vocabulary divided by the number of blocks as shown in Table 1.

Table 1. Tokens distribution over blocks.

Bit Block	Accuracy
00	This, am, weather, ...
01	was, attaching, today, ...
10	I, better an, Great, ...
11	than, NDA, ...

The next step is to use the LSTM model to generate the stegotext. In our proposed model, we used a LSTM autoencoder which is a neural network model for sequence data that depends on the encoder-decoder LSTM architecture. For a sequence of data, an encoder-decoder LSTM is designed to read, encode, decode, and reconstruct the input sequence. The model's performance is assessed by its accuracy in reproducing the original input sequence. Once the model reaches the desired level of performance in recreating the sequence, the decoder part can be removed and saved to be used for decoding the sequence into its original message.

The LSTM model can learn the intricate patterns in the temporal order of input sequences and utilize internal memory to retain and apply information across long input sequences.

In our proposed model, the encoder part starts by selecting one bit stream from the bit block B at a time. The LSTM then selects one token from the corresponding set of tokens for the chosen block. After reading in the entire token blocks sequence, the hidden state or output of this model represents an internal learned representation of the entire bit block sequence as a fixed-length vector.

Subsequently, the LSTM generates text that appears natural. For example, given the bit string "1000011011" and the token distribution over blacks in Table 1, the LSTM can form the partial sentence shown in Table 2.

Table 2. Example stegotext generation.

Bit Stream	10	00	01	10	11
Token	I	am	attaching	an	NDA

The output of the LSTM model is a fixed length vector that provides a representation of stegotext file that will be provided to the blockchain for storing.

Evidence Storing. The blockchain technology is used in the proposed model as a distributed database for evidence storing. Blockchain is a well-known technology to store data in a secure and decentralized manner. Decentralized blockchains are immutable, which means that the data entered is irreversible.

The data are stored in blocks linked together. When the block reaches its capacity, the data undergoes encryption through an algorithm, resulting in the generation of a hexadecimal number known as a hash. This hash is subsequently incorporated into the header of the next block and combined with the remaining data within that block. This process forms a sequence of interconnected blocks, creating a chain of information.

This phase involves adding the steganography file to the blockchain as a regular transaction. Adding evidence would be restricted to authorized users based on the blockchain type. For example, in a permissioned blockchain system for digital forensics, the authorized user could be law enforcement agencies, court systems or trusted third parties. This includes the verification process which is the mining in the blockchain to add the data into a block. A unique cryptographic hash value is created for the new block to identify it. The blocks contain information such as transaction (steganography file), nonce, target bit, difficulty, timestamp, previous block hash, current block hash, and block ID, etc., and the blocks are cryptographically verified and chained up to form an immutable chain which is the blockchain.

The pseudocode for the overall proposed approach is offered in Figure 2.

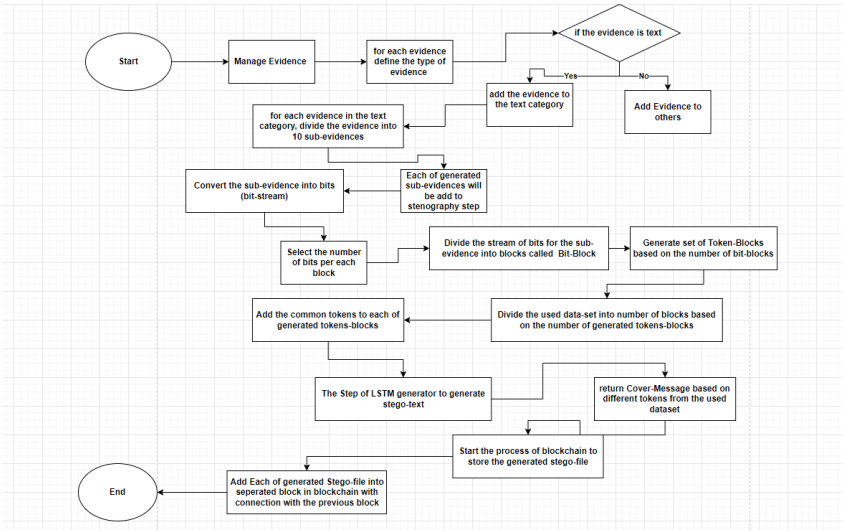

Figure 2. The proposed model.

4. Experimental Setup and Results

To implement the proposed model for creating cover files using an LSTM autoencoder, we emphasized the use of Python and its relevant libraries for neural network integration and evaluation. The work was implemented using a MacBook Pro with the following specifications: a 16-core M1 Max chip, 64 GB of unified memory, and a 4 TB SSD. This setup was chosen for its high-performance capabilities, allowing us to efficiently handle the computational demands of training and deploying our model.

4.1. Dataset

In this paper, the tiny_shakespeare dataset is used to test model performance [46]. The dataset consists of 40,000 lines of Shakespeare from a variety of Shakespeare's plays [46]. The tiny_shakespeare corpus is frequently used for training language models due to its manageable size and the complexity of Shakespeare's language. It strikes a good balance

between computational efficiency and the ability to produce intriguing text. The dataset is used to create vocabulary.

The dataset undergoes preprocessing, which includes tokenization. Tokenization is the process of breaking down long text strings into smaller units called tokens. Text can first be divided into sentences and then further into words. After that, redundant words are removed from the token set. The new set of unique tokens formed the vocabulary of our experiment and is then kept for use in the bit block mapping process.

Another dataset is used in this experiment for forensic evidences which is Email Communication Logs. The Email Communication Logs dataset is a collection of email exchanges between individuals, capturing the content, metadata, and structure of their communications. This dataset is particularly useful for research in forensic evidence analysis, social network analysis, and natural language processing [47]. The dataset contains the following features:

1. Sender and Recipient Information:
    - From: The email address of the sender.
    - To: The email address of the recipient(s).
2. Timestamp:
    - Date: The date and time when the email was sent.
3. Subject:
    - Subject: The subject line of the email, providing a summary of the email's content.
4. Email Body:
    - Body: The main content of the email, which includes the message text.

In this experiment, we utilized a subset of 100 samples from the dataset with the Body feature as the forensic evidence to be concealed and stored.

4.2. Experimental Setup

The experiment is conducted in several phases. In the first step, we examine the Email Communication Logs dataset to extract each email body as a piece of evidence. Each piece of evidence is divided into 10 chunks, with each chunk then converted into a sequence of bit streams to create bit stream blocks. These blocks are subsequently mapped to the 'vocabulary' tokens in the next phase.

The number of bits in each bit block plays a crucial role in the efficiency of the hiding process. A bit block represents the number of bits per block, meaning that the bit stream of the original secret message must be divided into blocks, each containing a specific number of bits. The purpose of defining this number of bits is to match the bit blocks with tokens in the token block. When the number of bits per block increases, the number of words required to construct the carrier message decreases.

Increasing the number of bits per block expands the number of available token blocks. This means that each token in the carrier message can represent more bits of the secret message. As a result, fewer tokens (or words) are needed to encode the entire secret message, effectively increasing the overall capacity.

Based on this, the selection of the number of bits per block has a significant impact on the capacity for information hiding. In one experiment, we tested our approach using different numbers of bits per block to observe the effects of using larger or smaller bit block sizes. The table below summarizes the results of this experiment.

In Table 3, "Bits per Block" refers to the number of bits chosen in each experiment to demonstrate the impact of using different bit block sizes. The "Number of Tokens per Block" column represents the calculated number of tokens in each block, determined by raising two to the power of the number of bits per block.

"Words Needed (Carrier Length)" indicates the number of words required in the carrier message to encode the entire secret message. This is calculated by dividing the total number of bits in the secret message by the number of bits each block represents.

Table 3. Effect of using a different number of bits per block representation.

Bits per Block	Number of Token per Block	Words Needed (Carrier Length)	Capacity
2	4	20	2
3	8	14	2.86
4	16	10	4
5	32	8	5
6	64	7	5.71
7	128	6	6.67
8	256	5	8
9	512	5	8
10	1024	4	10

Finally, "Capacity" refers to the number of bits that can be represented by a single word in the carrier message. The capacity is calculated by dividing the total number of bits in the bit stream by the number of words in the carrier message.

Based on the results in the table above, the capacity increases as the number of bits per block increases. This is because the capacity refers to representing each two bits in the secret message with one word in the carrier message. When the number of bits per block is increased, it becomes possible to represent four bits of the secret message with just one word in the carrier message, thereby enhancing the overall capacity.

This demonstrates that the proposed method offers a flexible solution, as the size of the carrier message can be controlled by adjusting the number of bits per block when constructing the bit block table.

In the next step of our experiment, we perform the mapping between the bit stream and the token block. Before doing so, we first determine the number of bits in the bit block; in this experiment, we chose three bits. Each bit sequence for each chunk of evidence is then divided by the number of bits in the bit block, resulting in multiple bit blocks for each chunk. Based on the number of bit blocks generated, we distribute the tokens over the bit blocks to create a token block for each bit block, ensuring that each token block has an equal number of tokens each token appears in only one token block.

To enhance the naturalness of sentences generated using the LSTM model, a set of common tokens will be added to all blocks of tokens. The purpose of these common tokens is to increase the naturalness of the sentences. These tokens will not convey any meaning for the secret message during the decoding and generation of the original secret message on the receiver's side.

After preparing our dataset of evidence and vocabulary, we established our LSTM autoencoder model. We defined the model architecture that expects a sequence of inputs equal to the number of bit blocks for each chunk and the number of features equal to the number of tokens in the token block. The output is a sequence with nine words and one feature. The model architecture consists of two LSTM layers, each with 100 units. We utilized Adam optimizer and trained the model for 100 epochs. Now our LSTM autoencoder will learn to reconstruct each input sequence. The effectiveness of the decoder ensures the integrity and security of the hidden information. The detailed steps of decoding are as follows:

1. Processing the Generated Text:
    - Tokenization: The generated text is tokenized into a sequence of tokens.
    - Embedding: Each token is converted into a dense vector representation using an embedding layer.
2. Re-encoding to Latent Space:
    - Sequence Encoding: The tokenized and embedded sequence is passed through an LSTM (or similar sequential model) to encode the sequence into a latent space

representation. This step ensures the decoder receives an input that matches the original encoding process.

3. Decoding the Latent Space:
   - Latent Space Input: The latent space representation is input into the decoder LSTM.
   - Bit Stream Generation: The decoder LSTM processes the latent representation to generate the output sequence, which is the recovered bit stream.
   - Bit-wise Output: The final layer of the decoder converts the decoder LSTM's outputs into bits, reconstructing the original bit stream.

We evaluated the performance of the decoder using the recovery rate metric.

4.3. Performance Metrics

To assess the effectiveness of our proposed model, we employ performance metrics, such as perplexity, which is a standard metric for the quality of language models, capacity, and coherence.

- Perplexity: Perplexity measures how well a probability distribution or probability model predicts a sample. In the context of language models, it quantifies how well a model predicts a given sequence of words. Mathematically, perplexity (PP) for a language model is calculated as:

$$\frac{1}{N}\sum_{i=1}^{N}\log P(w_i) \tag{1}$$

  where N is the number of words in the test set, and $P(w_i)$ is the probability assigned by the model to the i-th word in the test set. Lower perplexity values indicate better model performance in terms of prediction accuracy and language understanding.
- Capacity: The capacity refers to the number of bits that can be embedded or concealed within a single word of the carrier message. It can be defined as follows:

$$\frac{\text{Number of bits in the secret message}}{\text{Number of words in the carrier message}} \tag{2}$$

  In this equation, the number of bits in the secret message refers to the bits stream generated from the chunk evidence in the first step of the proposed solution, and the number of words in the carrier message refers to the words generated by the model used for cover file generation.
- Coherence: Coherence is a metric used to evaluate how logically or smoothly a piece of text flows. It assesses whether the sentences in a text are logically connected, making sense together in a cohesive manner. For language models like GPT-3.5, coherence can be evaluated based on factors such as sentence structure, logical flow, and clarity.

4.4. Experimental Results and Analysis

We selected three random sets of evidence from the Email Communication Logs dataset, each with varying sizes, to conduct our experiment. This random selection was made to ensure time efficiency and accommodate the limited resources of our laptop, which cannot process the entire dataset at once. After each run, we generated evaluation metrics to obtain the results. These results are presented in Table 4.

Table 4. Experiment results.

Subset	Subset Size	Perplexity	Capacity	Coherence	Accuracy
1	50-words	30	4	8/10	95
2	100-words	40	26	7/10	94
3	200-words	30	52	8/10	95

In Table 5, we present examples of words from the Email Communication Logs dataset and demonstrate how our model converts them into consistent natural language using an autoencoder.

Table 5. Example of generated natural language using an autoencoder.

Evidence to Be Hidden	Corresponding Stegotext in Natural Language
Hi Dave, just a reminder about our meeting scheduled for	Whether it's nobler in the mind to suffer the slings and arrows of outrageous fortune
Hello Smith	To be or not to be, that is the question
Thanks for the reminder, Carol	The lady doth protest too much, methinks

The breakdown of our proposed model is illustrated in the following example:

The first step in the proposed solution involves generating a bit stream based on the ASCII values of each character, as outlined below:

H: 01001000
E: 01000101
L: 01001100
L: 01001100
O: 01001111

Next, concatenate the bit representation of each character to form the following bit sequence: 0100100001000101010011000100110001001111.

To generate bit blocks, we use a block size of 3 bits for this experiment. Based on this distribution, the following bit blocks are created:

[010, 010, 000, 010, 001, 101, 010, 011, 000, 010, 011, 000, 010, 011, 111]

The token selected from the token blocks for each bit block is as follows:

000 → [“to”]
001 → [“be”]
010 → [“or”]
011 → [“not”]
100 → [“that”]
101 → [“is”]
111 → [“question”]
110 → [“the”]

The common tokens that used are as follows:

[“or”, “and”, “but”] [“be”, “are”, “is”] [“to”, “at”, “in”] [“not”, “no”, “none”] [“is”, “was”, “were”] [“question”, “query”, “inquiry”]

In our experiment, we employed an LSTM autoencoder to generate text from a bit stream of a secret message. The primary goal was to transform the bit stream into coherent and natural text while maintaining a high level of accuracy and efficiency. We evaluated the generated text using several metrics, including perplexity, capacity, and coherence, with the latter assessed using both a PERplexity Transformer (PERT) model and GPT-3.

First, we explored the effect of increasing the subset size on model performance. When we increase the subset size, we are essentially providing the model with more data to process. However, to manage this increased data efficiently, we fragment the text into smaller, manageable chunks. Each chunk is then processed independently by the model. This chunking approach is necessary to ensure that the model can handle large datasets without running into memory or computational limitations and to increase the security of our model as mentioned previously.

On the other hand, due to the fragmentation of the text, the model deals with each chunk separately, and the results for each chunk are then aggregated to form the final evaluation. This chunk-wise processing means that the model's performance on each individual chunk remains consistent, regardless of the overall subset size. As a result, the evaluation metrics, such as coherence, remain relatively similar whether we use a smaller or larger subset size.

Perplexity, as previously discussed, is a common measure in natural language processing used to evaluate language models. Our model achieved a low perplexity score, indicating that the LSTM autoencoder effectively converted the bit stream patterns and generated text that closely to the natural language. This low perplexity reflects the model's strong performance in understanding and encoding the input data. The LSTM autoencoder demonstrated high capacity by accurately capturing the complex patterns within the bit stream. This high capacity allowed the model to generate text that faithfully represented the encoded secret message, maintaining the integrity of the original information. The coherence of the generated text was evaluated using both the PERT model and GPT-3. Both models provided high coherence scores, affirming that the generated text was logically consistent and contextually appropriate. The high coherence scores suggest that the LSTM autoencoder produced text that was fluent and readable.

- PERT Model Evaluation: The PERT model assessed the logical flow and contextual relevance of the text, yielding high coherence scores. This indicates that the text generated by the LSTM autoencoder maintained a logical structure and stayed on topic.
- GPT-3 Evaluation: Using GPT-3 for coherence evaluation provided an additional layer of validation. GPT-3 can break down the text into tokens and compute the log probability for each token given the previous tokens. GPT-3's high coherence scores confirmed that the generated text was natural and human-like, further supporting the effectiveness of the LSTM autoencoder.

The values of coherence are heavily influenced by the perplexity score. A high perplexity indicates lower coherence, which in turn suggests that the generated text is less natural. Additionally, coherence is a subjective measure based on human evaluation, assessing whether the content of the generated carrier message is logically related and understandable.

In addition to generating text from a bit stream of a secret message, the LSTM autoencoder model can be used to reverse this process. Specifically, the decoder component can transform the generated text back into its original bit stream. To assess the performance of the decoder in this task, we used the recovery rate metric. This metric evaluates the percentage of original messages that were perfectly recovered without any errors. A high recovery rate indicates that the decoder effectively transforms the generated text back into the original bit stream with minimal errors, ensuring the integrity of the hidden message.

Our evaluation showed that the decoder achieved a high recovery rate of 96%, demonstrating the decoder's effectiveness in accurately reversing the encoding process.

Our analysis shows that the LSTM autoencoder is highly effective in generating text from a bit stream of a secret message. The model's low perplexity, high capacity, and excellent coherence scores highlight its ability to produce accurate, natural, and coherent text. These results demonstrate that the LSTM autoencoder can successfully transform bit streams into meaningful text, making it a valuable tool for text hiding.

Steganography is the practice of hiding secret information within an ordinary message to ensure that the presence of the secret information is not detectable. In our application, we used an LSTM autoencoder model to hide a secret bit stream by generating a corresponding natural text message. This method ensures that the hidden message is secure while the generated text appears natural and coherent to any observer. The model effectively hides secret messages within natural text, making it a powerful tool for steganographic applications. The generated text is coherent and natural, ensuring the hidden message remains undetectable. Additionally, the model demonstrates high accuracy in recovering the original bit stream, maintaining the integrity and security of the secret information. This approach combines the strengths of advanced language modeling and steganography, providing a secure and reliable method for covert communication.

The proposed steganography model addresses the need for enhanced security in digital communication and storage scenarios where confidentiality is paramount. It involves segmenting the data into chunks to ensure security, encoding the data to a bit stream and mapping the bit steam to a set of tokens, and finally using an LSTM autoencoder

to generate stegotext in natural language. This technique provide a robust solution for securely embedding and retrieving encrypted data within different types of carrier files that can be summarized as follows:

- Enhanced Security: The LSTM autoencoder generates natural and coherent text, making it indistinguishable from human-written text. This ensures that the presence of the hidden message is not easily detected.
- Complexity for Detection: By using an LSTM autoencoder, the generated text maintains a high level of coherence. This complexity in language modeling makes it difficult for detection algorithms to distinguish between regular text and text containing hidden messages.
- Adaptability: This approach can be adapted to different carrier file types and sizes while maintaining data confidentiality and integrity.

4.5. Comparison with Previous Works

To evaluate the robustness of our proposed solution in hiding evidence, we compared our approach against four other models proposed in the same domain of linguistic steganography to measure its performance and effectiveness using various datasets. These datasets were chosen to provide a thorough analysis of our model's performance across different types of text and to ensure that it is evaluated on multiple relevant dimensions. This comprehensive approach strengthens the validity of our experimental design and the robustness of our findings.

A brief description of each work and dataset is given below. In [48], the author proposed a text steganography method called GAN-TStega, which is based on Generative Adversarial Networks (GANs). While the Recurrent Neural Network (RNN) method has been used by [49] to propose a linguistic steganography. The authors in [50] proposed a multi-time-step-based steganography method for linguistic steganography. Lastly, the work presented in [45] introduces a new linguistic stegosystem based on a Long Short-Term Memory (LSTM) neural network. Below is a brief description of the four datasets used in the comparison experiment.

The Shakespeare Corpus is a compilation of texts from William Shakespeare's works, renowned for its linguistic richness and historical importance. This dataset is especially valuable for assessing natural language processing techniques within the realm of classical literature. Its complexity and diverse language structures offer a rigorous challenge for steganographic methods, pushing their ability to embed information while preserving textual coherence.

The News Headlines dataset offers modern, concise text data commonly used to evaluate text generation and summarization models. Its simple language and frequent repetitive patterns create both opportunities and challenges for steganographic methods, testing their ability to embed information while maintaining the natural flow of the text.

MSCOCO (Microsoft Common Objects in Context) dataset refers to a comprehensive collection of images and their corresponding captions, useful for evaluating models in multimodal contexts. Although primarily an image dataset, the associated captions provide textual data for steganographic evaluation.

IMDB which refers to the dataset that consists of movie reviews, which offer varied and dynamic language use, providing a challenging environment for steganographic methods. The dataset's diverse vocabulary and sentiment expression are critical for testing the effectiveness of text embedding techniques.

GAN-Stega utilizes Generative Adversarial Networks (GANs) to generate steganographic text. The generator is designed to create text capable of embedding hidden information, while the discriminator works to detect any hidden messages. For the implementation, the GAN model was trained on the same datasets (Shakespeare, News Headlines, MSCOCO, IMDB) using a comparable training process. The generator's role was to embed the secret message into natural-sounding text, while the discriminator measured detectability. Binary cross-entropy was used as the loss function for the discriminator, while the generator's loss

was a combination of adversarial loss and a perplexity-based loss to ensure text naturalness. In the bit-driven generator, the secret bits guided the word selection process, embedding the hidden information without sacrificing the naturalness of the text. The generated text was evaluated based on its security (the ability to conceal the secret message) and naturalness (linguistic quality). TensorFlow/Keras was employed for model implementation, with standard preprocessing techniques like tokenization and text normalization applied to prepare the datasets. By replicating the GAN architecture and training it on the same datasets, we ensured a fair comparison. The model's text generation capabilities were assessed using perplexity and other metrics from our model, providing a consistent benchmark against our proposed solution. RNN-Stega employs Recurrent Neural Networks (RNNs) to generate text that embeds hidden information within its structure. The RNN is trained to predict the next word in a sequence, with the hidden message guiding the word selection process. For the implementation, an RNN-based model was developed and trained on the same datasets (Shakespeare, News Headlines, MSCOCO, IMDB). During the text generation process, secret messages were embedded by controlling word selection at each step, influenced by the RNN's predictions and the bit stream. The model was built using TensorFlow/Keras, focusing on predicting the next word based on the previous context and the secret message bitstream. The loss function used in this approach combined traditional language modeling objectives with an additional objective to ensure successful embedding. To maintain consistency, the RNN architecture was aligned with the methodology described in the original paper. By using the same datasets, we ensured that the results were directly comparable, with the evaluation centered on metrics such as text naturalness, security, and detectability. The core of LSTM model in [45] is built on a Long Short-Term Memory (LSTM) network, a variant of Recurrent Neural Networks (RNNs) that excels in sequence generation tasks. The LSTM was trained on a large text corpus to predict the next word based on previous context, enabling it to learn a robust language model. To maintain consistency with other approaches, we applied the same preprocessing techniques, including tokenization and normalization. Each dataset was tokenized into words or sub-words, followed by normalization steps such as lowercasing, punctuation removal, and other adjustments to ensure uniform input for the LSTM model. A vocabulary was then created, reflecting the token diversity in each dataset. MSTS-Stega relies on a multi-time-step (MTS) approach, where perfect binary tree coding is used to map secret messages to word space. The language model is conditioned to generate text such that each word carries a portion of the hidden message. The secret message is divided into bits, which are mapped to a sequence of words using the binary tree coding structure. The language model is trained on the selected datasets, with additional steps to integrate the binary tree system, ensuring that each generated word represents part of the secret message. At each time step, a candidate pool of words is generated based on the language model's predictions. The pool size is constrained by the height of the binary tree. During text generation, candidate word combinations are evaluated using XOR operations on their binary codes. The combination that matches the secret bitstream and has the highest conditional probability is selected. The final steganographic text is formed by sequentially choosing and appending the appropriate words. In terms of evaluation, the generated text is assessed for its ability to effectively conceal the secret message. The same evaluation metrics, including perplexity, are used to measure the naturalness of the text, ensuring a consistent and accurate comparison.

Table 6 presents a comparison between various solutions across different datasets and evaluation criteria.

Various metrics are used to evaluate the effectiveness of the proposed linguistic steganography model compared to other approaches in the same domain. The results highlight key performance criteria, including perplexity, capacity, naturalness, security, and detectability, which provide a comprehensive evaluation of each method's strengths and limitations.

Table 6. A comparison between the proposed solution and various solutions across different datasets and evaluation criteria.

Criterion	Dataset	Proposed Solution	GAN TStega [48]	RNN Stega [49]	LSTM [45]	MTS-Stega [50]
Perplexity	Shakespeare Corpus	4.8	5.54	5.12	7.35	6.6
	News Headlines	5.0	5.54	5.12	7.35	6.60
	MSCOCO	4.7	5.6	5.6	5.55	5.84
	IMDB	4.9	4.7	4.76	4.9	5.21
Capacity Bits per block	Shakespeare Corpus	6	4	3	1	5
	News Headlines	6	4	3	2	4
	MSCOCO	6	4	3	2	5
	IMDB	5	4	3	2	5
Naturalness	Shakespeare Corpus	8.5/10	8/10	8.2/10	7.5/10	8.3/10
	News Headlines	8.3	8	8.2	7.7	8.3
	MSCOCO	8.4	8.1	8.3	7.6	8.2
	IMDB	8.6	8	8.1	7.4	8.1
Security	Shakespeare Corpus	False Positive Rate: 2%	False Positive Rate: 5%	False Positive Rate: 4%	False Positive Rate: 6%	False Positive Rate: 3%
	News Headlines	False Positive Rate: 2%	False Positive Rate: 5%	False Positive Rate: 4%	False Positive Rate: 6%	False Positive Rate: 3%
	MSCOCO	False Positive Rate: 2.1%	False Positive Rate: 4.8%	False Positive Rate: 2.3%	False Positive Rate: 5.2%	False Positive Rate: 3.3%
	IMDB	False Positive Rate: 2.2%	False Positive Rate: 3.8%	False Positive Rate: 2.1%	False Positive Rate: 2.7%	False Positive Rate: 4.3%
Detectability	Shakespeare Corpus	5% Detection Rate	10% Detection Rate	12% Detection Rate	15% Detection Rate	5% Detection Rate
	News Headlines	5% Detection Rate	8% Detection Rate	12% Detection Rate	13% Detection Rate	7% Detection Rate
	MSCOCO	6.3% Detection Rate	8.4% Detection Rate	10.9% Detection Rate	14.1% Detection Rate	7.6% Detection Rate
	IMDB	5.8% Detection Rate	8.6% Detection Rate	9.3% Detection Rate	11.9% Detection Rate	7.16% Detection Rate

Perplexity, a measure of prediction accuracy and text coherence, plays a critical role in assessing the quality of the generated text. The proposed solution achieves the lowest perplexity scores across most datasets, particularly excelling with the Shakespeare Corpus (4.8) and IMDB (4.9). These values reflect a strong language model that closely aligns with the original text structure, outperforming GAN-TStega and RNN-Stega. The lower perplexity of our model suggests that it generates more natural and coherent text compared to MTS-Stega, which shows higher perplexity values. This demonstrates that our model not only hides information efficiently but also maintains a high level of readability.

Capacity, a measure of the amount of information that can be embedded in a text, is another area where the proposed solution stands out. With a capacity of 6 bits/block in the Shakespeare Corpus and News Headlines, it outperforms other methods such as GAN-TStega and RNN-Stega, which achieve only 4 bits/token. This high capacity is crucial for scenarios requiring the embedding of larger amounts of hidden information without significantly altering the carrier message. Although MTS-Stega comes close, the proposed solution strikes a better balance between capacity and linguistic quality, as higher capacities in steganography often negatively affect the naturalness of the text. Our model achieves a high capacity while still generating readable, coherent text.

Naturalness, which is evaluated through human judgment, is critical for ensuring that the steganographic text does not arouse suspicion. The proposed model achieves a naturalness score of 8.5/10 for the Shakespeare Corpus and 8.6/10 for IMDB, slightly ahead of competing methods such as GAN-TStega and RNN-Stega. This indicates that our model successfully embeds hidden messages while maintaining high linguistic integrity. LSTM and MTS-Stega struggle more with naturalness, possibly due to their more rigid word selection processes and higher perplexity scores, which lead to less fluent or predictable text. By keeping the naturalness intact, our approach ensures that the stegotext remains indistinguishable from regular text, a key goal in steganography.

Security is evaluated by the false positive rate, with lower rates indicating a more secure system. Our model's false positive rate of 2% for both the Shakespeare Corpus and News Headlines reflects its strong resistance to detection, surpassing methods such as GAN-TStega and RNN-Stega, which exhibit higher false positive rates. The low rate ensures that legitimate text is less likely to be incorrectly flagged as containing hidden

information, enhancing the security of the steganographic system. Although MTS-Stega offers competitive security, the overall robustness of our approach, as seen across multiple datasets, reinforces the reliability of our solution in real-world scenarios.

Detectability, which measures how easily a steganographic method can be identified, is another area where the proposed solution excels. With a detection rate of only 5%, our model demonstrates strong resilience to detection mechanisms. This low detectability is critical in covert communication scenarios, where the goal is to ensure that the presence of hidden information goes unnoticed. While competing methods such as GAN-TStega and RNN-Stega exhibit higher detection rates, our approach minimizes detectability without compromising the quality of the generated text. LSTM shows the highest detection rate, likely due to more predictable patterns, while MTS-Stega, though competitive, still falls short of our model's performance. The findings from the evaluation demonstrate the superiority of the proposed linguistic steganography model across key metrics, highlighting its effectiveness and robustness. The model consistently outperforms other approaches in perplexity, achieving lower values that indicate more coherent and natural text generation, particularly excelling with datasets such as the Shakespeare Corpus and IMDB. Additionally, it offers a higher capacity for embedding hidden messages, managing to conceal more information per block without significantly compromising the text's naturalness, which remains competitive with top-rated methods. Furthermore, the model achieves excellent security, demonstrated by the lowest false positive rates, making it highly resistant to detection. Its detectability is also minimized, with lower detection rates compared to other models, underscoring its resilience in real-world applications where stealth is critical. Overall, the proposed solution strikes an optimal balance between capacity, text quality, and security, solidifying its position as a highly effective approach for linguistic steganography.

One notable drawback of our proposed approach is its reliance on extensive data preprocessing, including tokenization, normalization, and noise reduction. These preprocessing steps are time-consuming and computationally intensive. Moreover, the model's performance is sensitive to the quality and consistency of the preprocessing, which can impact its robustness and generalizability across different datasets. Despite these challenges, our model outperforms other models in the domain of linguistic steganography and demonstrates significant improvements in performance and effectiveness. This advantage highlights the trade-off between preprocessing complexity and the superior results achieved by our approach.

4.6. Potential Risks and Limitations

Despite the promising results of our proposed model, several potential risks and limitations must be considered for its application in real-world forensic investigations. These include scalability concerns, integration challenges, and other factors that may impact its practical deployment as listed below.

1. Scalability Issues: One significant concern is scalability. As the volume of digital evidence grows, the model's ability to handle large datasets efficiently may be challenged. The current implementation is designed to work with a manageable scale, but adapting it for extensive real-world applications will require addressing performance and processing constraints to ensure it remains effective and efficient.
2. Integration with Existing Systems: The model is proposed as a standalone solution and does not currently include detailed integration with existing forensic tools and frameworks. Implementing it within established forensic workflows may require additional adaptation and compatibility considerations.
3. Complexity of Implementation: The integration of steganography and blockchain components introduces complexity. Ensuring that these components work seamlessly together, while maintaining their respective security and performance standards, may pose challenges during implementation.

4. Future Technological Developments: The model's effectiveness is based on current technologies. Future advancements in areas such as quantum computing could impact the security and reliability of the blockchain component, necessitating ongoing updates and adjustments to address emerging threats.

We acknowledge these limitations and recognize the need for further research to address scalability and other challenges. Future work will focus on refining the model to enhance its practical applicability and integration with existing forensic practices.

5. Conclusions and Future Works

Preserving digital evidence involves more than just retention; it requires maintaining its credibility throughout the entire investigative process, from seizure to court presentation. Multi-layered security frameworks are essential for safeguarding evidence against unauthorized access and tampering, ensuring controlled sharing, and preserving evidence in a tamper-proof manner. The integration of blockchain technology offers a decentralized and immutable ledger, significantly enhancing evidence security, maintaining an unbreakable chain of custody, and facilitating traceability. Additionally, steganography provides a valuable method for securely transferring evidence by concealing it within innocuous files, reducing the risk of detection. Our proposed framework combines blockchain, steganography, and cryptography to strengthen the security, privacy, and authenticity of digital evidence.

Our analysis demonstrates that the LSTM autoencoder is highly effective in generating natural text from a secret bit stream, with low perplexity, high capacity, and excellent coherence scores highlighting its ability to produce accurate and coherent text. This effectiveness underscores the model's potential for text hiding applications, ensuring that the hidden message remains secure and undetectable while appearing natural to observers. The model's high accuracy in recovering the original bit stream further reinforces its reliability in preserving the integrity of the secret information.

Scalability presents a unique challenge in our framework, particularly with the use of language models. Our algorithm begins with a vocabulary set that includes all possible tokens appearing in the stegotext, which can be words or punctuation marks. To handle large datasets, tokens are divided into disjoint blocks, with each token appearing exactly once per block. The size of each block is determined by dividing the total vocabulary size by the number of blocks. As the dataset grows, managing the relationship between the number of bits and tokens becomes crucial. Larger datasets require more blocks, increasing the complexity and computational power needed for processing.

For future work, we intend to explore how different encoding strategies affect the model's performance and to investigate improvements to the autoencoder's architecture for enhanced efficiency and security. Additionally, we will evaluate the model's adaptability to various types of carrier files and languages to increase its applicability and robustness. We also plan to focus on optimizing token distribution and block management strategies to improve scalability while ensuring operational efficiency. Our aim is to refine these approaches to better manage computational load and tackle the complexities arising from larger datasets and more advanced language models.

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

Funding: This research was funded by the Deanship of Scientific Research at Birzeit University.

Data Availability Statement: All the data used in this research are an open source and available online.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Nakamoto, S.; Bitcoin, A. A peer-to-peer electronic cash system. *Bitcoin* **2008**, *4*, 15.
2. Sun, R.T.; Garimella, A.; Han, W.; Chang, H.L.; Shaw, M.J. Transformation of the transaction cost and the agency cost in an organization and the applicability of blockchain—A case study of Peer-to-Peer insurance. *Front. Blockchain* **2020**, *3*, 24. [CrossRef]
3. Hepp, T.; Wortner, P.; Schönhals, A.; Gipp, B. Securing physical assets on the blockchain: Linking a novel object identification concept with distributed ledgers. In Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems, Munich, Germany, 15 June 2018; pp. 60–65.
4. Surakhi, O.M.; AlKhanafseh, M.Y. Review on the Application of Blockchain Technology to Compact COVID-19 Pandemic. In Proceedings of the 2021 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 16–18 November 2021; pp. 193–198.
5. Tripathi, A.K.; Akul Krishnan, K.; Pandey, A.C. A Novel Blockchain and Internet of Things-Based Food Traceability System for Smart Cities. *Wirel. Pers. Commun.* **2023**, *129*, 2157–2180. [CrossRef] [PubMed]
6. Nam Nguyen, H.; Anh Tran, H.; Fowler, S.; Souihi, S. A survey of Blockchain technologies applied to software-defined networking: Research challenges and solutions. *IET Wirel. Sens. Syst.* **2021**, *11*, 233–247. [CrossRef]
7. Viriyasitavat, W.; Hoonsopon, D. Blockchain characteristics and consensus in modern business processes. *J. Ind. Inf. Integr.* **2019**, *13*, 32–39. [CrossRef]
8. Bamakan, S.M.H.; Motavali, A.; Bondarti, A.B. A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Syst. Appl.* **2020**, *154*, 113385. [CrossRef]
9. Sriman, B.; Ganesh Kumar, S.; Shamili, P. Blockchain technology: Consensus protocol proof of work and proof of stake. In *Intelligent Computing and Applications: Proceedings of ICICA 2019*; Springer: Singapore, 2021; pp. 395–406.
10. Majeed, U.; Khan, L.U.; Yaqoob, I.; Kazmi, S.A.; Salah, K.; Hong, C.S. Blockchain for IoT-based smart cities: Recent advances, requirements, and future challenges. *J. Netw. Comput. Appl.* **2021**, *181*, 103007. [CrossRef]
11. Ge, L.; Wang, J.; Zhang, G. Survey of consensus algorithms for proof of stake in blockchain. *Secur. Commun. Netw.* **2022**, *2022*, 28125263. [CrossRef]
12. Atlam, H.F.; Alenezi, A.; Alassafi, M.O.; Alshdadi, A.A.; Wills, G.B. Security, cybercrime and digital forensics for IoT. In *Principles of Internet of Things (IoT) Ecosystem: Insight Paradigm*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 551–577.
13. Selamat, S.R.; Yusof, R.; Sahib, S. Mapping process of digital forensic investigation framework. *Int. J. Comput. Sci. Netw. Secur.* **2008**, *8*, 163–169.
14. Englbrecht, L.; Pernul, G. A privacy-aware digital forensics investigation in enterprises. In Proceedings of the 15th International Conference on Availability, Reliability and Security, Virtual, 25–28 August 2020; pp. 1–10.
15. Javed, A.R.; Jalil, Z. Byte-level object identification for forensic investigation of digital images. In Proceedings of the 2020 IEEE International Conference on Cyber Warfare and Security (ICCWS), Islamabad, Pakistan, 20–21 October 2020; pp. 1–4.
16. Garfinkel, S.L. Digital forensics research: The next 10 years. *Digit. Investig.* **2010**, *7*, S64–S73. [CrossRef]
17. da Silveira, C.M.T.; de Sousa, R., Jr.; de Oliveira Albuquerque, R.; Amvame Nze, G.D.; de Oliveira Júnior, G.A.; Sandoval Orozco, A.L.; García Villalba, L.J. Methodology for forensics data reconstruction on mobile devices with Android operating system applying in-system programming and combination firmware. *Appl. Sci.* **2020**, *10*, 4231. [CrossRef]
18. Kumar, S. An emerging threat Fileless malware: A survey and research challenges. *Cybersecurity* **2020**, *3*, 1.
19. Palutke, R.; Block, F.; Reichenberger, P.; Stripeika, D. Hiding process memory via anti-forensic techniques. *Forensic Sci. Int. Digit. Investig.* **2020**, *33*, 301012. [CrossRef]
20. Hausknecht, K.; Foit, D.; Burić, J. RAM data significance in digital forensics. In Proceedings of the 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 25–29 May 2015; pp. 1372–1375.
21. Nelson, R.; Shukla, A.; Smith, C. Web Browser Forensics in Google Chrome, Mozilla Firefox, and the Tor Browser Bundle. In *Digital Forensic Education: An Experiential Learning Approach*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 219–241.
22. Ghafarian, A. An Empirical Analysis of Email Forensics Tools. Available online : https://ssrn.com/abstract=3624617 (accessed on 5 September 2024).
23. Javed, A.R.; Ur Rehman, S.; Khan, M.U.; Alazab, M.; Reddy, T. CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 1456–1466. [CrossRef]
24. AlKhanafseh, M.Y.; Surakhi, O.M. VANET Intrusion Investigation Based Forensics Technology: A New Framework. In Proceedings of the 2022 IEEE International Conference on Emerging Trends in Computing and Engineering Applications (ETCEA), Amman, Jordan, 16–18 November 2021; pp. 1–7.
25. Quan, Y.; Li, C.T.; Zhou, Y.; Li, L. Warwick image forensics dataset for device fingerprinting in multimedia forensics. In Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME), London, UK, 6–10 July 2020; pp. 1–6.
26. Yari, I.A.; Zargari, S. An overview and computer forensic challenges in image steganography. In Proceedings of the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 21–23 June 2017; pp. 360–364.

27. Barbier, M.; Le Bars, J.M.; Rosenberger, C. Image watermarking with biometric data for copyright protection. In Proceedings of the 2015 10th IEEE International Conference on Availability, Reliability and Security, Toulouse, France, 24-27 August 2015; pp. 618–625.
28. Muh, H.A.; Riadi, I. Analysis of Steganographic on Digital Evidence using General Computer Forensic Investigation Model Framework. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 315–323.
29. Majeed, M.A.; Sulaiman, R.; Shukur, Z.; Hasan, M.K. A review on text steganography techniques. *Mathematics* **2021**, *9*, 2829. [CrossRef]
30. Karampidis, K.; Kavallieratou, E.; Papadourakis, G. A review of image steganalysis techniques for digital forensics. *J. Inf. Secur. Appl.* **2018**, *40*, 217–235. [CrossRef]
31. Kamal, R.; Hemdan, E.E.D.; El-Fishway, N. Forensics chain for evidence preservation system: An evidence preservation forensics framework for internet of things-based smart city security using blockchain. *Concurr. Comput. Pract. Exp.* **2022**, *34*, E7062. [CrossRef]
32. Abulaish, M.; Haldar, N.A.H.; Jahiruddin, J. P2DF: A Privacy-Preserving Digital Forensics Framework. *Int. J. Digit. Crime Forensics* **2021**, *13*, 1–15. [CrossRef]
33. Verma, R.; Govindaraj, J.; Gupta, G. DF 2.0: Designing an automated, privacy preserving, and efficient digital forensic framework. In Proceedings of the Annual ADFSL Conference on Digital Forensics, Security and Law, Las Vegas, NV, USA, 27–28 May 2018.
34. Malik, A.; Sharma, A.K. Blockchain-based digital chain of custody multimedia evidence preservation framework for internet-of-things. *J. Inf. Secur. Appl.* **2023**, *77*, 103579.
35. Brotsis, S.; Kolokotronis, N.; Limniotis, K.; Shiaeles, S.; Kavallieros, D.; Bellini, E.; Pavué, C. Blockchain solutions for forensic evidence preservation in IoT environments. In Proceedings of the 2019 IEEE Conference on Network Softwarization (NetSoft), Paris, France, 24–28 June 2019; pp. 110–114.
36. Li, S.; Qin, T.; Min, G. Blockchain-based digital forensics investigation framework in the internet of things and social systems. *IEEE Trans. Comput. Soc. Syst.* **2019**, *6*, 1433–1441. [CrossRef]
37. Akbarfam, A.J.; Heidaripour, M.; Maleki, H.; Dorai, G.; Agrawal, G. ForensiBlock: A Provenance-Driven Blockchain Framework for Data Forensics and Auditability. *arXiv* **2023**, arXiv:2308.03927.
38. Wan, X.; He, J.; Huang, N.; Mai, Y. Ontology-Based Privacy Preserving Digital Forensics Framework. *Int. J. Secur. Appl.* **2015**, *9*, 53–62. [CrossRef]
39. Alruwaili, F.F. Custodyblock: A distributed chain of custody evidence framework. *Information* **2021**, *12*, 88. [CrossRef]
40. Anitha, C.; Priyadharshini, R.; Sivajothi, E.; Kumaran, G.; Sudha, K.; Sireesha, B. Experimental Evaluation of Secured Forensic Evidence Handling Scheme using Blockchain Technology. In Proceedings of the 2023 4th IEEE International Conference on Intelligent Technologies (CONIT), Bangalore, India, 21–23 June 2024; pp. 1–6.
41. Menard, T.; Abouyoussef, M. Towards Privacy-Preserving Vehicle Digital Forensics: A Blockchain Approach. In Proceedings of the 2024 12th IEEE International Symposium on Digital Forensics and Security (ISDFS), San Antonio, TX, USA, 29–30 April 2024; pp. 1–6.
42. Karras, A.; Karras, C.; Schizas, N.; Avlonitis, M.; Sioutas, S. Automl with bayesian optimizations for big data management. *Information* **2023**, *14*, 223. [CrossRef]
43. Tageldin, L.; Venter, H. Machine-Learning Forensics: State of the Art in the Use of Machine-Learning Techniques for Digital Forensic Investigations within Smart Environments. *Appl. Sci.* **2023**, *13*, 10169. [CrossRef]
44. Hema, A.M.; Kuppusamy, K. A novel trust-based privacy preservation framework for service handling via ontology service ranking. *Wirel. Pers. Commun.* **2020**, *112*, 1339–1354. [CrossRef]
45. Fang, T.; Jaggi, M.; Argyraki, K. Generating steganographic text with LSTMs. *arXiv* **2017**, arXiv:1705.10742.
46. Tiny Shakespeare Dataset. Available online: https://raw.githubusercontent.com/karpathy/char-rnn/master/data/tinyshakespeare/input.txt (accessed on 9 August 2024).
47. Enron Email Dataset. Available online: https://www.cs.cmu.edu/~enron/ (accessed on 9 August 2024).
48. Yang, Z.; Wei, N.; Liu, Q.; Huang, Y.; Zhang, Y. GAN-TStega: Text steganography based on generative adversarial networks. In Proceedings of the Digital Forensics and Watermarking: 18th International Workshop, IWDW 2019, Chengdu, China, 2–4 November 2019; Revised Selected Papers; Springer: Berlin/Heidelberg, Germany, 2020; pp. 18–31.
49. Yang, Z.L.; Guo, X.Q.; Chen, Z.M.; Huang, Y.F.; Zhang, Y.J. RNN-stega: Linguistic steganography based on recurrent neural networks. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 1280–1295. [CrossRef]
50. Yu, L.; Lu, Y.; Yan, X.; Yu, Y. Mts-stega: Linguistic steganography based on multi-time-step. *Entropy* **2022**, *24*, 585. [CrossRef]