

APPLYING MACHINE LEARNING TO PREDICT SYMMETRIC ENCRYPTION ALGORITHM INPUTS

A Thesis Presented to

The Faculty of the Computer Science Department

California State University Channel Islands

In (Partial) Fulfillment

of the Requirements for the Degree

Masters of Science

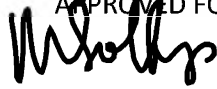
Report Compiled by: Kelly Armstrong

Advisor: Dr. Michael Soltys

MSCS Graduate 2019-2021

June, 2021

APPROVED FOR MS IN COMPUTER SCIENCE



Oct 26, 2021

Dr. Michael Soltys (Advisor)

Date



Reza Abdolee (Nov 5, 2021 12:08 PDT)

11/05/2021

Dr. Reza Abdolee

Date



11/05/2021

Dr. Ivona Grzegorzczuk

Date

APPROVED FOR THE UNIVERSITY



Jill Leafstedt (Nov 8, 2021 16:27 PST)

11/05/2021

DR. Jill Leafstedt

Date

Non-Exclusive Distribution License

In order for California State University Channel Islands (CSUCI) to reproduce, translate and distribute your submission worldwide through the CSUCI Institutional Repository, your agreement to the following terms is necessary. The author(s) retain any copyright currently on the item as well as the ability to submit the item to publishers or other repositories.

By signing and submitting this license, you (the author(s) or copyright owner) grants to CSUCI the nonexclusive right to reproduce, translate (as defined below), and/or distribute your submission (including the abstract) worldwide in print and electronic format and in any medium, including but not limited to audio or video.

You agree that CSUCI may, without changing the content, translate the submission to any medium or format for the purpose of preservation.

You also agree that CSUCI may keep more than one copy of this submission for purposes of security, backup and preservation.

You represent that the submission is your original work, and that you have the right to grant the rights contained in this license. You also represent that your submission does not, to the best of your knowledge, infringe upon anyone's copyright. You also represent and warrant that the submission contains no libelous or other unlawful matter and makes no improper invasion of the privacy of any other person.

If the submission contains material for which you do not hold copyright, you represent that you have obtained the unrestricted permission of the copyright owner to grant CSUCI the rights required by this license, and that such third party owned material is clearly identified and acknowledged within the text or content of the submission. You take full responsibility to obtain permission to use any material that is not your own. This permission must be granted to you before you sign this form.

IF THE SUBMISSION IS BASED UPON WORK THAT HAS BEEN SPONSORED OR SUPPORTED BY AN AGENCY OR ORGANIZATION OTHER THAN CSUCI, YOU REPRESENT THAT YOU HAVE FULFILLED ANY RIGHT OF REVIEW OR OTHER OBLIGATIONS REQUIRED BY SUCH CONTRACT OR AGREEMENT.

The CSUCI Institutional Repository will clearly identify your name(s) as the author(s) or owner(s) of the submission, and will not make any alteration, other than as allowed by this license, to your submission.

APPLYING MACHINE LEARNING TO PREDICT SYMMETRIC ENCRYPTION ALGORITHM INPUTS

Title of Item

Computer Science Master Thesis

3 to 5 keywords or phrases to describe the item

Kelly Armstrong

Author(s) Name (Print)

Kelly Armstrong
Kelly Armstrong (Nov 5, 2021 11:33 PDT)

Author(s) Signature

11/05/2021

Date

APPLYING MACHINE LEARNING TO PREDICT SYMMETRIC ENCRYPTION ALGORITHM INPUTS

Kelly Armstrong

October 19, 2021

Abstract

The motivation for the topic of this thesis is to use Machine Learning to reverse engineer hash functions. Hash functions are supposed to be hard to reverse one-way functions. The Machine Learning algorithm will learn the hash function with a probability above 50 percent which means we can improve our guess of the inverse. This is done by implementing the DES symmetric encryption function to generate N many values of DES with a set key and the Machine Learning algorithm is taught a neural network to recognize the first bit of the input based on the value of the function's output. A new table is created and used for testing where the new table is created similarly but has different inputs. The Machine Learning algorithm, XGBoost using scikit-learn, runs on the new table and compares it to the other table and a confusion matrix is used to measure the quality of the guesses.

Contents

1	Introduction	1
1.1	Motivation and Contributions	1
1.2	Literature review	2
2	Background	4
2.1	Parity	4
2.2	DES as a one way hash function and objectives in cybersecurity	4
2.3	Hash Functions and Cybersecurity, Cryptographic Hash functions, One-way functions	10
2.4	DES in Cryptography	13
2.5	Key based cryptography	15
2.6	DES Algorithm	15
2.7	Cybersecurity and cloud computing	16
2.8	The importance of Symmetric and Asymmetric Encryption	18
2.9	Weak and Strong end-to-end Encryption	23

2.10	Artificial Intelligence, Neural Networks and Machine Learning Algorithms . .	28
2.11	Gradient Boost, gradient descent, Boosting	31
2.12	XGBoost Algorithm	33
2.13	Stochastic Cyber Attacks	35
2.14	Confusion Matrix	36
3	Contribution	37
3.1	Reviewing introductions and stating contributions	37
3.2	Description of Contributions	39
3.3	Snippets of code and Flowchart of Algorithm	42
3.4	How DES is used to contribute to cybersecurity	52
3.5	How Machine Learning is used to contribute to cybersecurity	54
3.6	Better than a coin toss	56
3.7	Description in detail of contributing to security	58
3.8	Cloud Encryption and Cloud Security	59
4	Experiments and Justifications	62
4.1	Stochastic Cyber Attack	62

4.2	DES weak encryption standard	63
4.3	Using ML to break DES Encryption	64
4.4	Running the experiment	65
4.5	Algorithm Description and explaining the experiment	67
4.6	pydes.py	69
4.7	Do.py	70
4.8	Generate.py, nextpart.py	70
4.9	xg.py	70
4.10	Value of using XGBoost	71
4.11	DES encryption standard as weak algorithm	72
4.12	Guessing first bit	73
4.13	Confusion matrix from three test runs	73
4.14	Decision tree plot	76
5	Conclusion and future work	80
	References	83

List of Figures

1	Flowchart of Algorithm with code	51
2	DES Diagram	53
3	close up of decision tree nodes and branches	76
4	close up of decision tree nodes and branches	77
5	close up of decision tree nodes and branches	77
6	close up of decision tree nodes and branches	78
7	Decision tree corresponding to Figure 15	78
8	close up of decision tree nodes and branches	79
9	close up of decision tree nodes and branches	79

1 Introduction

1.1 Motivation and Contributions

The motivation of this thesis is to show that a machine learning algorithm can be applied to predict the first bit of input that has been encrypted with a one-way function, which should not be able to be reversed or guessed. If the machine learning algorithm can guess the input (or preimage) of a one-way function without knowing the contents of the input (or preimage) then the function is reversible which means the function is not secure for encryption because it can be reversed. If the function can be reversed, then the contents can be exposed and so it is not secure. Hash functions are defined as one-way functions. If a machine learning algorithm can reverse a secure hashing encryption algorithm it can guess the secret key and reveal the contents of the document or message.

The contribution of this thesis will show that with a probability of greater than 50 percent the input of a symmetric hash function, DES, can be learned by a supervised machine learning algorithm, XGBoost, hence demonstrating that there exists a higher possibility for an attack against the input (or preimage) of a symmetric hash function which by definition should be collision resistant. The digested message of a hash function should not be possible to reverse engineer. With machine learning techniques it is possible to guess the first bit of encrypted input which means it is possible to apply similar techniques to reverse engineer an entire message. The details containing the contribution of this thesis are located in section 3.2 on page 42.

1.2 Literature review

Most researchers just use a brute force attack or guess and check approach when it comes to predicting passwords. The same is likewise true for criminals and devious hackers. These methods can be time consuming and may never return a correct guess in a reasonable amount of time. There has not existed a successful algorithm using machine learning that can predict an encrypted document since most methods just use brute force attacks which are usually unsuccessful if the encryption methods are strong. So, other methods need to be discovered.

Other research techniques include the example of a timestamp that encodes a specific time that data was accessed. Keeping encryption secure to protect time sensitive information is a major concern in areas of cyber security. Their idea of constructing a black box guess to unlock a piece of data, called a timelock puzzle[13], using a random oracle model to construct an input has been tested extensively to produce a unique input. [13] A random oracle model is a controversial topic in cryptography because a complete proof of a security has not been solved by researchers. A random oracle model is a theoretical black box construction that is like a hash function because it resembles the concept that the output is random given a unique input. A theoretical black box outputs a unique response from its output domain that is unique to each query or random oracle model will always input that is submitted. A random oracle model will always respond in the same way for any unique input. Each response is chosen uniformly, so it is continuous on the interval at each point in the interval and the output, even though it is arbitrary, is bounded. This research technique is broad and offers a wide range of applications.

Other methods use rule based password guessing or Markov Models. Other researchers have used deep learning to predict passwords from PassGAN "to train a neural network to determine autonomously password characteristics and structures, and to leverage this knowledge to generate new samples that follow the same distribution." [85] In this research method the neural network is being trained to determine characteristics and structure of a password. This research was effective and included guessing the password as a whole string of bits and did not focus on the predictability of one bit of correctly guessed data.

Machine learning techniques can be applied to predict encrypted data and a more detailed specific approach could be needed in this field. Using XGBoost to learn the first bit of data that has been encrypted with DES is an example of a specific problem and will be investigated in this thesis. This research will narrow down a technique that can be applied to other problems in this field and focus on a specific encryption method using a specific machine learning algorithm to predict, successfully, the first bit of input data.

2 Background

The significance behind the contribution of this thesis is to use machine learning to break encryption methods without having a password or decryption key and without finding a flaw in the hardware or algorithm. Using machine learning techniques to predict encrypted inputs will alter encryption standards in practice. Future research can apply the results in this thesis to prepare for security practices in the future of cloud computing.

2.1 Parity

In Boolean algebra the word parity is used to determine if a value is 1 if and only if the input vector has an odd number of ones. In Computer parity means something slightly different, it is used to describe the evenness or oddness of the number of bits with the value of 1 when there is a set of bits, and the value is then determined by all the bits. It can be assumed that the calculation is done using an XOR summation of the bits that gives 0 or 1. A value of 0 will be even and a value of 1 will be odd.

2.2 DES as a one way hash function and objectives in cybersecurity

In the 2004 movie "I, Robot" a detective must put his trust in a robot to help save humanity from evil robots who have decided to overthrow the authority of mankind for some

reason. “There have always been ghosts in the machine. Random segments of code, that have grouped together to form unexpected protocols. Unanticipated, these free radicals engender questions of free will, creativity, and even the nature of what we might call the soul. Why is it that when some robots are left in darkness, they will seek out the light?” [36] Although this is a science fiction interpretation of artificial intelligence it certainly can be used metaphorically to think about how machine learning applications could be used for the prevention of malicious attacks or alternatively be used as a form of a malicious cyber-attack. If we can use machine learning to decode encryption methods, there is no telling how this can affect the cyber security industry for good or for worse. Methods of cybersecurity are critical to protecting privacy and public safety. “Cybersecurity is generally understood to be a set of techniques and measures taken to protect digital information against unauthorized access or attack.” [1]

There is a difference between encryption which is a two way function that transforms plaintext into ciphertext since it can be decrypted and a hashing function is strictly one way. However, DES can be used as a one way hash function if it has a good enough block cipher. A good block cipher is considered good if the keyspace of a cipher is large enough. The larger keyspace makes it complicated to break the key using brute force attacks. If the keyspace in DES is large enough it could be considered good enough to be a one-way function. DES takes plaintext and through a series of permutations and XOR functions it transforms the plaintext into ciphertext. Hashing gives a input of plaintext a hash value, which can be accomplished using DES. “DES is the best known and most widely used encryption function in the commercial world today. Generating a one-way hash function which

is secure if DES is a “good” block cipher would therefore be useful.”[86]

Before a signature is applied the message is digested with a hash. The hash function maps data of some random or arbitrary size to a fixed size and the digest is the output of the hash function which is called a hash. Hashing is a process that converts a known key into another key. ”It is usually achieved through data encryption, either with symmetric or asymmetric (i.e., public key) algorithms.”[1] We do not want our data to be able to be accessed by outside parties and modified or disrupted in anyway thus preserving the integrity of the message or document.

It has been discussed in literature what the direct objectives of cybersecurity are. “Objectives of cybersecurity: confidentiality, integrity, availability, authentication, authorization, accounting, and non-repudiation. “[1] We want methods that encrypt our documents to be unbreakable. In other words we need to be able to trust encryption standards to maintain security and privacy. We need to trust that these methods of cybersecurity are confidential which is defined in literature “to prevent the disclosure of information to unauthorized entities (people or systems).

Objectives of cybersecurity can be summed up using the following definitions from literature.

These objectives are:

- Confidentiality Confidentiality means the data is kept private and other users are not able to access that data unless the original person who owns or is sending the data

allows access.

- **Integrity** Integrity means that the data is not able to be modified. “Integrity: data cannot be modified undetectably.”[1] A useful application how to preserve the integrity of the document can be using a digital signature or even a timestamp. “Another measure for integrity is digital signatures (also known as digital digests) which both authenticate a document and ensure its integrity. This is achieved with hashing functions and public key cryptography.”[1] This technique uses the reverse of public key use and for confidentiality you encrypt with a secret key but decrypt with public key but in signatures you sign with your public key. Your signature authenticates the document but also can protect the integrity, you know it is not changed in transit. You hash the document, and you encrypt the hash with your secret key proving you are in possession of it but your public key is known to the world so anyone can verify the signature. Public key is used both for confidentiality and integrity but in reverse order.
- **Availability:** Availability which is important because it includes protecting against human error or natural disaster. “Availability: information is available when it is needed, that is, both the computer system, and the communication channels are functioning correctly.”[1] Availability points to an important aspect of cybersecurity because you are not only protecting against malicious attackers, but you are protecting from accidents or faults in a system.
- **Authentication:** It is important to know the document has not been tampered with and that the data is original and unaltered. Authentication ensures that the data is genuine. “Authentication: to ensure that data, transactions, communications or documents are

genuine.”[1] Authorization stipulates what this agent is allowed to do.[1]

- Authorization: Authorization uses credentials to verify actions or privileges to access data.
- Accounting: Another important aspect of cyber security is to ensure that a document is authentic and to keep track of the data through accounting. Accounting keeps record logs, and automating parsing them and reacting in near real-time.[1]
- Non-repudiation: An alternative for understanding accounting can be explained with non-repudiation. “Non-repudiation: implies that one party of a transaction cannot deny having received a transaction nor can the other party deny having sent a transaction; an example of application would be online bidding. Note that non-repudiation can also be seen under Accounting”[1]

Even though the study of cryptography is a fancier topic than security, cryptography is still a small subset of cybersecurity albeit an important one. “Security groups known as firewalls filter network traffic, and LoadBalancing, which help in the scaling of demand and discussed in the section on “Availability,” are not cryptographic applications.”[1] It is important to know the difference between cryptographic applications and other categories in cybersecurity. Some topics relate specifically to the security of protecting the hardware of cyber systems and may not be considered cryptographic applications at all. Since cryptography is a subset of cybersecurity it is important to discuss critical concepts that focus on its elegance. Hashing is an example of a one-way function that is used in cryptography. One-way functions are important and is critical concept in cryptography and security. Using

machine learning algorithms for encryption problems is a reasonable problem in today's research areas concerning cybersecurity. There are many machine learning algorithms available and it is dependent on the type of data available. "XGBoost dominates structured or tabular datasets on classification and regression predictive modeling problems." [2] For problems that include tabular data XGBoost is a highly effective and efficient algorithm. This particular problem of using machine learning to guess the input of encrypted data is a predictive modeling problem. "Predictive modeling is a method of predicting future outcomes by using data modeling." [33] Once a pattern is observed for how machine learning algorithms are able to predict inputs further work can be done to develop models on stronger encryption methods. Predictive modeling is a good approach to this type of problem and in particular using machine learning to make accurate predictions is a preferred technique. "Unlike statistics, where models are used to understand data, predictive modeling is laser focused on developing models that make the most accurate predictions at the expense of explaining why predictions are made. Unlike the broader field of machine learning that could feasibly be used with data in any format, predictive modeling is primarily focused on tabular data." [27]

XGBoost is an efficient boosting algorithm. Boosting is a numerical optimization strategy and it is a problem that minimizes loss by the addition of weak learners. Weak learners are added in repetition to the model. "The idea of boosting came out of the idea of whether a weak learner can be modified to become better." [2] The goal is to use weak learners in a greedy algorithm that makes for an efficient learning model by turning a poor hypotheses into a good hypotheses. XGBoost is the perfect algorithm for efficiency and can solve tasks quickly, it is known as "...an efficient algorithm for converting relatively poor hypotheses into

very good hypotheses.”[2] Using a weak learning method can improve performance, and our strategy is to achieve better than a random guess or random chance. ”A weak hypothesis or weak learner is defined as one whose performance is at least slightly better than random chance.”[2] XGBoost is the algorithm of choice since it is a highly efficient and fast machine learning algorithm.

2.3 Hash Functions and Cybersecurity, Cryptographic Hash functions, One-way functions

Law enforcement uses techniques in cryptography to crack codes or decrypt messages from criminal offenders, this application of cryptography is called cryptanalysis. Forensic investigators can utilize cryptanalysis to try and break the encryption that a hacker may have used when launching an attack. This is done by using a mathematical algorithm that attempts to crack the code, or cipher, that was used to encrypt data. Once a cipher has been determined it can be used to discover evidence which can be used to prosecute criminals. “A cipher is an algorithm for performing encryption or decryption”. [19] Cryptography tactics such as these help law enforcement agencies crack codes, protect data from being stolen, gather evidence against criminal offenders, protect data, and authenticate access for users of a system. There are many applications of cryptography including using cryptographic hash functions to encrypt important data or messages. Cryptography plays a role in many applications in cybersecurity. The mathematics behind cryptography is elegant and useful to many cyber security professionals. Cryptography applications and tools assist in digital

forensics and solving cybercrimes. However, cyber criminals can also use cryptography for their own malicious intentions and even hide traces of crimes they commit or tamper with evidence. Cybersecurity professionals use encryption methods such as hashing to protect sensitive information and data from being changed or altered by outside malicious hackers, since exposing this information may cause crucial ramifications in a judicial process.

Hashing is an algorithm that transforms a string of some input or string of data into a shorter value of fixed length that is a transformed version or key of the original input. “Hashing is the practice of using an algorithm to map data of any size to a fixed length.” [20] Hashing allows for the fast retrieval of data from a shorter hashed key value in comparison to a longer string of data that the hash is based on. “Hashing is used to index and retrieve items in a database because it is faster to find the item using the shorter hashed key than its original value.” [20] Hashing is used in cryptography since it transforms some input into some smaller unreadable output. “Hashing is a method of cryptography that converts any form of data into a unique string of text. Regardless of the data’s size, type, or length the hash that any data produces is always the same length.” [23]

Hashing is considered a secure method used to encrypt data , but it depends on the type of hash function that is being used. The hash value is scrambled from its original input through a cryptography algorithm. The output is a unique value corresponding to a numerical hash value. The hash value is unique to the input of the original document or the original contents of the data. The hash value of the output is specific to the input, it can only be altered if the original input is altered. “A hash function is any function that can be used to map data of arbitrary size to fixed size values, usually used to index a fixed

size table called a hash table.” [20] DES is an example of a weak hashing algorithm and will be discussed in detail in section 2.9. An example of a strong hash function is AES-256. “AES-hash is a secure hash function, meaning it takes an arbitrary bit string as input and returns a fixed length (in this case, 256 bit) string as output. Any alteration of the input should completely garble the output.” [60] There are many different types of hash functions and hashing algorithms. Some hash functions are distinguished from other hash functions by having certain properties that others do not.

A cryptographic hash function is a hash function that is a mathematical algorithm that maps some input of some bit size and takes the input of some amount of data and outputs a fixed size of a string of bytes called a hash value. Every cryptographic hash function is a hash function but not every hash function is cryptographic. [23] The algorithm is a one-way function and is not possible to reverse. One-way functions are irreversible functions hence the name “one way” implying they only work in out direction and an inverse is not possible. “It’s easier for a hacker to bypass cryptography by exploiting a vulnerability in the system than it is to break the mathematics.” [22] One-way functions are useful in cryptography. Encryption involves being able to invert or decrypt the original input but hashing is only one directional. “Encryption is a two-way function and hashing is a one way function.” [21] One-way functions are impossible to reverse and do not have inverses. “A one way function in computer science is a function that is easy to compute on every input but hard to invert given the image of random input.” [18]

One-way functions provide extra security since encryption involves being able to invert a message using a key, if that key becomes intercepted then the contents of the data

become compromised. One-way functions are secure since it is not possible to decode the contents back to the original input once they are encrypted through the hash function. “A cryptographic hash function guarantees security of properties. Non-cryptographic hash functions just try to avoid collisions for non-malicious input.” [23] Cryptographic Hash function is a particular type of hashing algorithm. “A cryptographic hash function is an algorithm that takes an arbitrary amount of data input and produces a fixed size output of enciphered text called a hash.”[23]

2.4 DES in Cryptography

The DES algorithm is the most popular encryption algorithm to this date. Since DES is so popular and so well known it has been hacked several times and is no longer considered secure. DES is a weak encryption algorithm that is useful for research examples. DES is a 56-bit symmetric encryption algorithm that stands for Data Encryption Standard. “DES is a 56 bit key symmetric algorithm the same key is used for encryption and decryption. DES is a symmetric key block cipher created in 1970 by IBM adopted by NIST.”[64] The algorithm takes the plain text in 64 bit blocks and converts them into ciphertext using 48-bit keys. Blocks cipher algorithm takes plaintext of 64 bits converts to ciphertext using keys of 48 bits. DES has 16 rounds.[63,64]

DES is not considered secure its 56 bit key is a short length. AES data encryption is more mathematically efficient for longer key lengths such as 128 bit, 192 bit or 256 bit making it exponentially stronger than 56-bit key of DES. 3-DES is better than DES because

it applies DES algorithm three times. The AES is the current strongest encryption symmetric key algorithm.

DES is a symmetric key algorithm that is 56 bits and works by encrypting a 64 bit message or piece of data by using encryption keys that are also 64 bits long where each 8th bit is ignored creating a key size of 56 bits. "Use of a 56-bit key is one of the most controversial aspects of DES. Even before DES was adopted, people outside of the intelligence community complained that 56-bits provided inadequate security"[81] Stronger encryption methods use longer key lengths which makes it harder to decrypt, but DES has been used widely in cryptography. This experiment was done using DES in the hopes of advancing further applications in cryptography and cyber security. [76] A key is what is used to decrypt or decode the output of a cryptographic function or algorithm and corresponds to a unique transformation of plain text and converts it into cipher text.[76] If a machine learning technique can be used to decrypt messages without a key that it is not a secure encryption method. It is like locking the door to a house that has no walls. A key would be rendered useless if the output itself becomes easy to predict.

Encryption and decryption can be used reciprocally in that a cipher can be used for encryption and decryption or a new key can be required for the decryption algorithm or the reverse encryption algorithm. An encryption key is a random string of bits that is unique to the specific data that is being encrypted and it is created to decode data. Encryption keys are created to be intentionally unique and not able to be predicted. If a key is very long then it is hard to decode the encryption. An encryption key looks different to humans than it does to "robots" or computers who are only able to see data in terms of 0's and 1's. XOR

is a function that happens everywhere in encryption which can be connected to modular polynomial arithmetic which over a finite field uses Galois field theory which is essentially a lot of long divisions and additions over a finite field. [76]

2.5 Key based cryptography

Key based cryptography:

1. Divide plain text
2. Implement DES on individual blocks
3. Combine all blocks
4. Blocks become cipher
5. Complete DES algorithm
6. Permutations provide transformation of Keys
7. Key transformed to completion

In DES the key length is 56-bits but it is not always the case that a key is this length, infact DES is criticised for having too short of a key length. The DES algorithm will be discussed in the next sections.

2.6 DES Algorithm

DES has sixteen rounds of operations, and each DES round has the same operations and uses a different round key.[64] The permutation works by changing the position of the bits

and each DES round will take a ciphertext as input that is produced by the previous round and will output ciphertext that will be used as input for the next round after it is divided into the left and right halves. The mangler function takes 32 bit and expands it to 48 bits, then the 48 key substitutes into the 32-bit value. [64]

This can be expressed mathematical by the following formulas:

$$L_{n+1} = R_n \tag{1}$$

$$R_{n+1} = L_n XOR(R_n, K_n) \tag{2}$$

2.7 Cybersecurity and cloud computing

Cybersecurity is the protection of data and connected systems from cyber threats or cyber terrorism. “Cybersecurity is generally understood to be a set of techniques and measures taken to protect digital information against unauthorized access or attack.” [1] Cloud computing definition: the practice of using a network of remote servers hosted on the internet to store, manage, and process data, rather than a local server or a personal computer. [80]The

Cloud has undoubtedly become a significant resource for anyone who needs to be able to store mass amounts of data. "Cloud computing is both flourishing and evolving. "[80] The Cloud could also become a sustainable resource for law enforcement however there are justified concerns regarding the security of moving the location of data pertaining to national security or matters of state evidence.

A concern for law enforcement may be storing evidence or any classified information inside of a cloud-based network. Many laws prevent local law enforcement from moving evidence from onsite locations. To use a cloud computing network, it is essential that cybersecurity measures of the utmost importance are considered. For example, it would be necessary to prove that hacking or evidence tampering has not occurred. If there is any reasonable doubt that the evidence is not genuine or if there is any suspicion of evidence tampering or any security breaches, then it may weaken a case in criminal proceedings or leave the nation vulnerable to an enemy attack. Therefore, to argue in favor of using the Cloud for matters of law enforcement or national security storage it is necessary to ensure that an encryption scheme for Cloud computing is put into place. Methods of encryption and decryption need to be strong, secure, and unable to decode. "Encryption is the process of translating plaintext data into something that appears to be random and meaningless (ciphertext). Decryption is the process of converting ciphertext back to plaintext. To encrypt more than a small amount of data symmetric encryption is used." [45]

There needs to be a strong encryption algorithm that can be adopted uniquely by law enforcement agencies to avoid hacking or evidence tampering. To argue in favor of using cloud storage it is necessary to ensure that an encryption scheme over a cloud network can

exist that is unbreakable hence un-hackable.

2.8 The importance of Symmetric and Asymmetric Encryption

There is a significant difference between encryption and decryption. It is important to know the difference because it is an essential part of cyber security. “Cybersecurity is the protection of internet connected systems and data from cyberthreats”. [40] Encryption is a procedure that includes protecting the authenticity of the document or data by using an algorithm to scramble its contents so that it is unreadable to whoever does not have permission to access the document or data. “To encrypt convert a message or document or any type of data into a cipher or a code. This is to prevent the access from someone who is not supposed to view the data or document or whatever information. It conceals the information by converting it into some unreadable code.” [40] Decryption is the method that involves reading that “un-readable code”. It interprets the message to the reader by unscrambling the encryption method. “Decryption is the process of converting ciphertext back to plaintext, to encrypt more than a small amount of data then symmetric encryption is used.” [43] Encryption uses an algorithm to scramble data so that it is unreadable, and decryptions uses the reverse to unscramble the data so that it is readable. Encryption uses an algorithm to scramble (or encrypt) data and then uses a key to unscramble or decrypt the information. [47]

Encryption methods are a fundamental tool to applications of cryptography. Encryption provides security and protects communication methods. Encryption and decryption are tools needed in cybersecurity practices especially over applications and servers used in

data or message transfer. “Encryption and decryption occur in the application layer.”[48]

Encryption protects the data or document that is being sent by verifying keys from the user that is sending the message or data through an application or server by using methods to convert the data into unreadable code so that it cannot be understood by anyone without a key. “Data encryption translates data into another form or code so that only people with access to a secret key formally called a decryption key or a password can read. Encrypted data is ciphertext and encrypted data is plaintext. The decryption key is a code you need to transform an encrypted message or document or data into a form that is then able to be read.” [40] Encryption is necessary for matters of security in military or law enforcement channels since the information can continue to contain sensitive or confidential information. “Similar to the encryption process the document to the decryption input and the decrypted result is the output. In encrypted unreadable form it is ciphertext.”[46] Unless there is a way of decoding the ciphertext it is not possible to understand the intention of the message or to decode the data. It is important to choose a secure method of encryption so that the data cannot be easily decoded. Without a secure and trusted encryption method matters of security could be compromised. In order to decrypt data or a document the person receiving it from the sender is able to decode the data or document with a private key. An example is an email message being sent including a pdf document that has a passcode. The receiver or the email message should know the passcode to open the document. The passcode is an example of a private key since it should only be shared between the sender and receiver of the message where the recipients email address is the public key since it is known to the public. The difference between public key vs. private key and choosing the right type of encryption methods is a common question that arises. “A private key is used to both encrypt

and decrypt data and is shared between the sender and receiver of encrypted data and to decrypt the data the private key is used and is shared, the public key is free to use and the private key is kept secret.” [50]

It is important to use strong encryption methods. There have been many secure encryption standards over the years. Encryption techniques include: AES, DES, RSA, triple, blowfish, two fish, and others.[51] It is not possible to decrypt something without a key or without knowledge of flaws in the system being used. “Unless there is a flaw in the algorithm and you know it. The only option is brute force, and it may take hundreds or years.” [47] It is theoretically possible that even encrypted data could be compromised. If there is infinite time and resourcing, it is possible that encrypted data can be decrypted to reveal original content. Hackers rely on flaws in the system or resort to other means of hacking attacks at attempts to decrypt messages. While a brute force approach may work it is not necessarily probable considering the amount of computing resources that would be required and the time that it may take to do so. Hackers rely on other decryption methods to attack a system or application to intercept data or messages. “The most common way to hack encrypted data is to add an encryption layer using an attacker’s key. To decrypted messages or documents you need the private key to which the message is encrypted.”[46]

There are several methods of encryption that also have several types of styles. One example is the distinction between symmetric vs asymmetric encryption. “Symmetric encryption uses one key for both encryption and decryption and asymmetric uses public key for encryption and private key for decryption.” [44] There is a noticeable difference between symmetric and asymmetric algorithms. Even so there are differences in the types of

symmetric algorithms. “There are two type of symmetric encryption algorithms. The block algorithm which uses a set length of bits are encrypted in blocks on electronic data with the use of a specific secret key. The stream algorithm is encrypted as it streams instead of being retained in system memory. [41] The difference between symmetric and asymmetric encryption is an important topic since the choice between the two involves the type of task that is being asked to complete. Symmetric and asymmetric algorithms are different in many ways. There are several advantages to using either a symmetric or an asymmetric algorithm, it depends on what is trying to be done. Symmetric algorithms have problems with key exchange. In order to trust both the sender and the receiver of the encrypted item they each share a secret key. There could arise a sense of paranoia between how well either side can be trusted with the secret key or even how well secure either side is able to keep their application or server. This could cause problems in authenticity and integrity. Symmetric key algorithms in cryptography use the same keys, that could be identical or transformed, for encryption and decryption of plaintext and ciphertext respectively.

Asymmetric key algorithms also use a key algorithm that uses a key and plaintext inputted into a key algorithm that encrypts the data or message and outputs the ciphertext. The encrypted message uses the public key paired with the private key which is used in decryption. The one who receives the message uses their public key paired with their private key. There are benefits and disadvantages of both symmetric and asymmetric algorithms. Asymmetric encryption is more secure since only the receiver uses a paired public and private key. Symmetric encryption uses a single key that has to be shared by the receiver and the sender and the single key will decrypt the message. An example of a weak encryption is

DES which is used in cryptography and a more secure version of DES is called 3- DES or AES which is also used in network security and other cryptographic applications.

The question becomes what type of encryption method is most secure. In comparison between the types of symmetric encryption methods as well as the asymmetric encryption methods. It depends on the application or serve that is being used and also the type of message or data is being encryption. Many pros and cons can be found on several methods of encryption, and it depends on which is more important. Asymmetric encryption is more secure but symmetric encryption is much faster. “Asymmetric encryption is more secure while symmetric is faster. They are both effective it just depends on the task required. Main advantage of symmetric encryption would be it is fast and efficient for large amounts of data and the disadvantage is the need to keep the key secret.”[49] The task of assigning what encryption methods to what type of data can become complicated if many different factors are involved. “Challenging where encryption and decryption take place in difference places requiring the key to be moved. Symmetric encryption is faster because keys used are much shorter than in asymmetric encryption.” [49]

2.9 Weak and Strong end-to-end Encryption

There are many examples of weak and strong encryption methods. Even over the years what was once considered a secure encryption algorithm like DES has since been proven that it can be easily hacked and is not secure. DES used to be a standard encryption algorithm and is now considered a weak encryption algorithm. “A weak hash function cannot guarantee data integrity. A good hashing algorithm makes it difficult to forcibly generate two messages that have the same hash value.” [23] Choosing a weak hash function compromises data integrity and the overall intention of the document.

End to end encryption is a communication encryption tactic that includes symmetric and asymmetric encryption and is necessary in Cloud security or other encryption platforms. Examples of end to end encryption are found in symmetric and asymmetric encryption such as Elliptic curve cryptography which is symmetric and AES which is symmetric. Eliminating end to end encryption means that clients would rely on servers for encryption and would not have access to their own personal encryption keys.

A weak hash function could mean disaster if the data was intercepted and altered, so a weak hash function can not be trusted. When cybersecurity professionals are choosing a function for their encryption schemes it is necessary to consider several properties that make a hash function a strong one. “Strong hash functions have properties that makes it impossible to deliberately find collisions.” [57] A collision implies that two inputs have produced the same output and it could then be compromised since it would be easier to revert back to the original message if it is the case that a collision occurs. Hence a good cryptographic hash

function must have no collisions or hard to find collisions. “A cryptographic hash function provides three properties, well defined in the world of cryptography: collision resistance, pre-image resistance and second pre-image resistance.” [56] It is sufficient to say that a hash function should be collision resistant hence is resistant to preimage attacks. “Preimage resistance is the property of a hash function that it is hard to invert, that is, given an element in the range of a hash function, it should be computationally infeasible to find an input that maps to that element.” [62]

Preimage resistance is the property of the hash function that implies it is not possible to invert. If a hacker is easily able to invert the hash function, then the security of the function is compromised. If a message is altered or changed because it was able to be decoded, then the integrity of the message is no longer able to be trusted. It is necessary that hash functions have the security properties that make it resistant to these types of attacks. “Preimage attack on cryptographic hash functions tries to find a message that has a specific hash value. A cryptographic hash function should resist attacks on its pre-image.” [23]

Definition 2.1 (Preimage). For a given function, the set of all elements of the domain that are mapped into a given subset of the codomain; [12,18]

Collision resistant functions are resistant to preimage attacks. Collision resistance is a critical property in cryptographic hash functions. A strong hash function should be collision resistant. “Collision resistant is a security notion of cryptographic hash functions. A collision of a hash function is a pair of different inputs which give the same output. Weak collision resistance means the probability of failing to find a collision.”[34] A collision is when different inputs are able to give the same output, a secure function has weak collision resistance since a secure function would not have the property that different inputs are producing the same output. Second preimage resistance is when a second input has the same output as some other input. A secure function would also not share this property and would not have the same output as some other input. “Second preimage resistance is the property of a hash function that it is computationally infeasible to find any second input that has the same output as a given input.”[62] Since a function who has preimage resistance inherits the property that it also has second preimage resistance and also since a function who has collision resistance also inherits preimage resistance we can say that a collision resistant function also inherits the property of being second preimage resistance.

If a function has the property of collision resistance, then it is second preimage resistant. “Collision resistance implies second preimage resistance.” [23] If a function is collision resistant it retains the other properties and is resistant to all preimage attacks. A one-way function is a function that is non-reversible such that it is not possible to invert so combining these properties strengthen a hash function. “A collision free hash function is a

one-way function that is also collision resistant.” [34] A good choice for a hash function is one that is collision resistant and is also a one-way function that is resistant to all pre-image attacks since their guarantees integrity.

The strongest encryption available today is AES “AES, the Advanced Encryption Standard, is the algorithm trusted as the standard by the United States government and numerous organizations. It is extremely efficient in 128-bit form, AES also uses keys of 192-bits and 256-bits for heavy duty encryption purposes.”[53] AES-256 is not “crack able” since it is nearly impossible to decrypt or crack using any type of brute force method. “AES has never been cracked and is safe against any brute force attacks.” [55] If AES was simple to crack or if there was a way to break its encryption standard essentially the entire world and all of our secrets would be exposed. Essentially if it were the case that a method was able to decrypt AES none of the world’s secrets would be safe since nothing would be secure, it would be up for grabs and any information would be available for public eyes. “While a 56-bit DES key can be cracked in less than a day AES would take billions of years to break using current computing technology.[55]

Differences between cracking AES-128 algorithm and AES-256 algorithm are not that extreme and require less calculations respectively but is still ridiculously impossible to crack using brute force attacks. “If you’re using a 128-bit AES cipher, if a quantum system has to crack a 256-bit key, it would take about as much time as a conventional computer to crack a 128-bit key.” [53] There are other examples of encryption algorithms that are used and the choice depends on the data or task that is being performed. The RSA encryption algorithm is a great example of a strong encryption method because of the key lengths it

is able to protect. It supports incredibly long key lengths (typical to see 2048- and 4096-bit keys.[23] Calculating long prime numbers is computationally intense and there does not exist a known factoring method for simplifying the factorization of large primes. RSA is a powerful encryption standard that is much slower than AES and is typically used to encrypt small data sets since it is computationally intense.

DES is considered to be a weak encryption standard and is not typically used in practices today. 256-bit encryption is incredibly strong and arguably the strongest type of encryption available. 256-bits is the key length of the encryption function. AES-256 is the strongest encryption standard since it supports a bit size that is un-hackable by brute force attacks. “Using a GPU processor that tries 10.3 billion hashes per second cracking a password, that is longer with varied characters, would take approximately 507 billion years, which is approximately 37 times the age of the universe.” [54] Examples of strong encryption algorithm are ones that have 256-bit encryption schemes like AES-256 or SHA-256. “SHA-256 stands for Secure Hash Algorithm 256-bit and it’s used for cryptographic security. Cryptographic hash algorithms produce irreversible and unique hashes.” [61] Currently, there does not exist a plausible technology platform that has the capability to break 256-bit encryption algorithms. Many leading tech companies use SHA-256 to encrypt their servers. Amazon Cloud Services encrypt EC2 instances using SHA-256 in the AWS environment. [35]. It would literally take possibly billions of years to do so if the public tried with their own devices and even still the most sophisticated computing powers would also require a ridiculously long amount of time to break these encryption algorithms.

Is it impossible to break something that is encrypted with something as strong as AES-256? A good question arises from this if we assume it is possible then we have to imagine how long it would take for this to take place. “How long does it take to hack AES-256?” [51] So theoretically even if it was possible, we can assume that anything is possible, it is not something that would ever be able to occur in our life span or perhaps even in the life span of all mankind or even the universe itself. “There are 984,665,640,564,039,457,584,007,913,129,639,936 possible combinations. No supercomputer on this earth can crack that in any reasonable timeframe. Even if you use the fastest supercomputer available in the world it would take millions of years to crack a 256-bit AES encryption.” [52]

2.10 Artificial Intelligence, Neural Networks and Machine Learning Algorithms

Artificial Intelligence can be considered a controversial topic especially if we remember the dramatic interpretation of how many robots are portrayed in the media. The unpopular opinion that artificial intelligence can benefit mankind is something that could be shared among people who value growing with technology instead of evolving away from it. A popular movie references makes the example of getting rid of the internet to preserve libraries. “I suppose your father lost his job to a robot. I don’t know, maybe you would have simply banned the Internet to keep the libraries open.” [36] While this may be an extreme example and from a science fiction movie none the less it is still an interesting point that can be

made. So, while technology evolves if we hope to evolve with it we certainly need to be able to protect ourself from malicious attackers or from criminals who seek to use technology for malicious purposes. Encryption schemes are important to humans to preserve privacy and protect sensitive data. What would it mean if “robots” or artificial intelligence agents were able to decrypt our most powerful security methods? Cybersecurity will have to continue to evolve to support these possibilities. Artificial intelligence is a large field and machine learning is a subset of the topic of artificial intelligence. “AI is a bigger concept to create intelligent machines that can simulate human thinking capability and behavior, whereas, machine learning is an application or subset of AI that allows machines to learn from data without being programmed explicitly.” [66] Artificial Intelligence is a wide topic and includes several branches of computer science and mathematics. Machine learning is a fascinating subsection of the topic of artificial intelligence.

There are significant differences between the two topics even though machine learning fits into the category of artificial intelligence. “Machine learning is a subfield of artificial intelligence, which enables machines to learn from past data or experiences without being explicitly programmed.” [66] Machine learning and artificial intelligence are strongly related topics but they do have significant topics that do not intersect. Artificial intelligence is typically used to assign machines to complete tasks that have been done by humans. It is the study of intelligence gained and developed by machines. Machine learning is a process that trains a machine to learn from its environment or a large set of data interpreted as an input. Machines in machine learning are not programmed but instead learn from the environment of their problem to complete tasks. There are two types of learning that is done by machines

and they are called supervised and unsupervised learning. “Within artificial intelligence (AI) and machine learning, there are two basic approaches: supervised learning and unsupervised learning. The main difference is one uses labeled data to help predict outcomes, while the other does not.” [67] Supervised learning accomplishes tasks by using patterns in the system from training data. Unsupervised learning is a self-learner where the environment of a system is investigated to discover features from the input without any previously programmed set of categorical data. “Unsupervised learning is a machine learning technique, where you do not need to supervise the model. Supervised learning allows you to collect data or produce a data output from the previous experience. Unsupervised machine learning helps you to find all kinds of unknown patterns in data.” [68]

Deep learning is another topic in artificial intelligence where the machine is actually able to train itself. “neural networks make up the backbone of deep learning algorithms.” [77] It is different from machine learning methods that allow machines to learn from the systems environment and input data. Deep learning needs neural networks to accomplish being able to learn from the data and train itself to learn without human help. A neural network is a network of algorithms modeling itself off the human brain. It includes multiple networks of algorithms that discover familiar patterns and recognize relationships in a set of data through a process that mimics the way the human brain operates. [18] In the contents of deep learning, it is necessary to discuss what an artificial neuron is.

An artificial neuron is basically a mathematical function that receives one or more than one input then sums that input and produces an output. It was created to model biological neurons and is modeled based on the brains of the neural networks of the human or

other animal nervous systems.[18] The signal is a real number, and the output is computed by a non-linear function from the summation of the inputs. The signal is found at a connection and the connections are called the edges which have a weight. The weight will adjust as the learning progresses. [18]

There are many different types of neural networks available for problem solving. The k-nearest neighbor or KNN algorithm is a machine learning algorithm that is used in classification and regression problem. It uses data to classify new data from the similarity on the measurement of the distance from each neighboring node to each using a distance function for classification. A convolutional neural network or CNN is used in image recognition software to take input images and assign learnable weights on the image to distinguish between other images. CNN uses feature parameters to reduce computations and reduces dimensionality.

2.11 Gradient Boost, gradient descent, Boosting

Gradient boosting optimizes a loss function then adds decision trees, which are used as weak learners, sequentially, one after the other, and the previous learners are not changed. The weak learners make predictions, and the model continues to add weak learners so that it minimizes the loss function. The type of loss function is determined by the type of problem. The decision trees are greedy which means that it makes a locally optimal decision at each stage. Regression trees are used by binary recursive partitioning. “A regression tree is built through a process known as binary recursive partitioning, which is an iterative process that splits the data into partitions or branches, and then continues splitting each partition into

smaller groups as the method moves up each branch.” [38] The regression trees output real values at the split then the output is added together which creates the next model, then this processes repeats itself to split that model after the output is added together and creates the next model and so on. The added outputs are used for predictions.

Gradient boosting is a machine learning greedy algorithm that is used for optimizing predictive models through sequential steps as the algorithm learns. It builds an ensemble of weak learners that uses a loss function to make predictions to minimize error by over-fitting a training data set quickly. “A novel gradient boosting framework is proposed where shallow neural networks are employed as “weak learners”. General loss functions are considered under this unified framework with specific examples presented for classification, regression, and learning to rank.”[78] Gradient boosting uses weak learners as its neural network. To reduce over-fitting regularization should be used on certain parts of the algorithm, depending on the problem being solved. Each gradient boosting fits the model, tunes the parameters and makes predictions. Each sequence minimizes the loss function in the negative gradient and descends as each weak learner, the decision trees, is added until the best prediction is made. Each prediction model is minimized when it is combined with the previous model to optimize the predicted value of the model in each step of the learning process. It is an ensemble of weak models and is a supervised machine learning technique used to minimize loss functions in regression, ranking and classification problems. Boosting is a useful ensemble method that uses many algorithms for better predictability. Ensemble methods are when there are multiple algorithms used together.

Newer and newer models are added one after the other in sequence until no better

prediction is possible and each of the newer models correct the mistake from the last predictive models. The last model perfects the score of all other models and is the final addition to the sequence. A Gradient boosting algorithm uses models that are added together to make a final prediction and each new model is made such that each new model predicts the error of the last model. Gradient boosting incorporates a gradient descent algorithm that minimizes the loss to support regression and classification. Gradient descent is typically used in function space where gradient boosting uses decision trees, in a type of “block structure to support the parallelization of tree construction.” [2] The idea is to minimize the loss when each tree is added and minimize each set of parameters used. Examples of what could causes loss are a wide variety of parameters that include weights in a neural network or coefficients in a regression equation or weak learners like decision trees. Each error in the parameters is calculated and then updated to improve performance of the next model. Continue training so that you can further boost and already fitted model on new data [2]

2.12 XGBoost Algorithm

XGBoost is a popular gradient boosting algorithm that stands for “extreme gradient boosting. It has been designed intentionally to be a fast algorithm where it outperforms other algorithms “dominates structured or tabular datasets on classification and regression predictive modeling problems” [2] XGBoost is called extreme because it is an extreme gradient boosting algorithm that is meant to be computationally efficient and outperform all other computations for boosting tree algorithms. This algorithm is known for its execution speed and model performance. Boosting is a numerical optimization problem that minimizes the

loss of the model by adding weak learners and is called a stagewise additive model. The first boosting algorithm was called Adaptive Boosting which involves decision trees with a single split and weights the observations by putting difficult instances with more weights than new weaker learners are added sequentially. New weak learners are added sequentially that focus their training on more difficult patterns. [2]

The XGBoost algorithm is an efficient implementation of a greedy boosting algorithm where the library is using the gradient boosting decision tree algorithm but is designed for speed and accuracy. “Gradient boosting is a greedy algorithm and can overfit a training dataset quickly. It can benefit from regularization methods that penalize various parts of the algorithm and generally improve the performance of the algorithm by reducing overfitting.” [2] Regularization helps with overfitting by estimating the mean or median of the data to improve accuracy and speed. Regularization is a technique used for tuning the function by adding an additional penalty term in the error function. The additional term controls the excessively fluctuating function such that the coefficients don’t take extreme values. [32]

XGBoost is considered by many to be the best machine learning algorithm available. XGBoost is a machine learning algorithm that can run in the AWS cloud. [2, 35] The library has a gradient boosting implementation which is designed for quickly solving regression, ranking, prediction, and classification problems. XGBoost is a stochastic gradient boosting algorithm which is an effective and efficient algorithm that is specifically designed for high performance using tabular data that offers a high range of parameters. It uses multiple decision trees to predict the error and correct the error or residual from the last tree. This result comes from the loss function by trying to predict the target of the last tree

to minimize the predicted error. “XGBoost uses 2nd order derivatives as an approximation and advanced regulation which improves model generalization.” [39]

2.13 Stochastic Cyber Attacks

Machine learning algorithms typically behave stochastically. The behavior and performance of many machine learning algorithms is a variable process where the outcome involves a degree of random chance or general randomness. This property of randomness is why we refer to this behavior as being stochastic. “A variable or process is stochastic if there is uncertainty or randomness involved in the outcomes.”[59] In mathematics when probability theory is discussed it is often described as a random process or event where something random is happening to a mathematical object. Usually these mathematical objects are called random variables. “Stochastic processes are widely used as mathematical models of systems and phenomena that appear to vary in a random manner. randomly determined; having a random probability distribution or pattern that may be analyzed statistically but may not be predicted precisely.”[59]

A stochastic attack is a random chance of guessing the input of some encrypted message or data set. The random probability that a machine learning algorithm could accurately guess the input of an encrypted data set is the purpose of this project and is an example of a stochastic attack using a machine learning algorithm.

2.14 Confusion Matrix

Confusion matrix for machine learning is a matrix that compares the target values with the predicted values, where the predicted values are from the machine learning model. [58] The confusion matrix describes the performance of the algorithm. It is a table that is used to describe the performance of a classification model by calculating correct guesses and incorrect guesses. It is a specific table corresponding to specific values so that the performance is something that can be understood by human readers.[31]

Classification models use a set of testing data where the actual values are known. The classification model is on a set of test data, the confusion matrix makes the results easier to be interpreted.[31] It involves a ratio of the correct or true positives and a ratio of correct or true negatives and compares it to a ratio of incorrect or false positives and a ratio of incorrect or false negatives.

3 Contribution

3.1 Reviewing introductions and stating contributions

We have shown then that there is a stochastic attack on a hash function that can be slightly better than 50 percent chance which means it is slightly better than a guess. A probability 0.01 percent over 50 percent means we can predict the next bit of encrypted input using machine learning. DES is a weak encryption standard, and the first bit of encrypted data can be reversed engineered with a machine learning algorithm.

The contribution of this thesis is to use XGBoost to predict the first bit of data encrypted with DES. The purpose of this thesis is to examine how machine learning can help with reverse engineering hash functions which are assumed to be hard to reverse. Artificial Intelligence has been the subject of science fiction movies for several generations. It is slowly starting to become a more political debate than it is a matter of science fiction with emerging technologies. Artificial Intelligence is a highly advanced topic and has a wide range of applications. Machine learning is a subset of Artificial Intelligence where it is an application of artificial intelligence[65] Machine learning is what happens when machines learn from some form of data without being programmed in detail to do so or to perform a learned task. If we use machine learning to our advantage it becomes more of a question of ethics than it is a question of integrity. We can use machine learning to crack the secrets of the universe including our own passwords.

The motivation of this thesis is to show that a machine learning algorithm can be applied to break a one-way function, which should not be able to be reversed or guessed. If the machine learning algorithm can guess the input (or preimage) of a one-way function without knowing the contents of the input (or preimage) then the function is reversible which means the function is not secure for encryption because it can be reversed. If the function can be reversed, then the contents can be exposed and so it is not secure. Hash functions are defined as one-way functions. If a machine learning algorithm can reverse a secure hashing encryption algorithm it can guess the secret key and reveal the contents of the document or message. The contribution of this thesis will show that with a probability of greater than 50 percent the input of a symmetric hash function, DES, can be learned by an supervised machine learning algorithm, XGBoost, hence demonstrating that there exists a higher possibility for an attack against the input (or pre-image) of a symmetric hash function which by definition should not be able to be reversed. Since a hash function should be collision resistant the preimage should be not able to be guessed or learned, however through this thesis it will be shown that this is, in fact, possible as well as probable. A question arises: How to choose the right machine learning algorithm and which encryption standard to use.

3.2 Description of Contributions

Any accuracy score above 50 percent means the algorithm was able to learn a pattern for correctly predicting the encrypted bit of data. It means that the algorithm successfully guessed the correct bit and learned how to guess correctly. Several experiments using these algorithms has predicted the first bit of encrypted input data with a higher accuracy over 50 percent. Exact figures of the accuracy score are showed in section 4.13 and show the output of the printed accuracy score from three runs of the algorithm. We can guess the nest bit using machine learning and further apply this tactic to continue to predict each bit of encrypted data. This experiment uses machine learning to reverse engineer a hash function. Hash functions are assumed to be impossible to invert one way functions and are resistant to preimage attacks. It has been shown that a machine learning algorithm can predict the first bit of encrypted data which a higher accuracy than 50 percent which means that the first bit of the input of a DES encrypted input can be predicted with a higher accuracy than guessing. Using a weak encryption standard such as DES gives insight into how machine learning can be used to break encryption standards. By using machine learning to predict the first bit of encrypted data we can examine how with the help of machine learning it is possible to reverse engineer hash functions by means of a stochastic attack.

When researchers in the past have discussed the differences in unsupervised learning and supervised learning it involves a distinction in how data is collected. In unsupervised learning techniques you do not need a model that is supervised, and this is considered a machine learning technique since the algorithm is used to to learn from input data. Supervised

learning allows the collection of data to be used to make a determination. Supervised learning produces a data output based on how it has been able to deduce the data from previous learning experiments. Unsupervised learning allows for the detection of patterns in unknown data.

The contributions to the current research in cybersecurity are that it is possible to use a machine learning algorithm to successfully guess encrypted data. Specifically, the first bit of encrypted data from a weak encryption function. What it might mean to connect DES symmetric encryption to be connecting to machine learning. It might mean that any security proof would not be possible since no theoretical black box construction could be completed since any unique input would be decrypted by being chosen uniformly. “What if, no matter how strong your password was, a hacker could crack it just as easily as you can type it? In fact, what if all sorts of puzzles we thought were hard turned out to be easy? Mathematicians call this problem P vs. NP, it is perhaps the single most important question in computer science today.” [77]

Cryptosystems are built from the ground up on hash functions. Uniform distribution implies that an output is chosen arbitrary, but the value is totally bounded and continuous on the interval at each point in the interval. The output domain is chosen uniformly but each output is unique. This could imply that it is susceptible to a pre-image attack because it is not collision resistant if the input is able to be regurgitated. The input would only have to exist and would not have to be unique to the encrypted data.[69,72] These attacks imply that the choice of hash function would be irrelevant. Continuous functions can fail a uniform continuity test if there are points in the continuous function that

are not continuous. The output domain of a random oracle model follow that the output domain follows uniform distribution and is uniformly chosen at random arbitrary outcome between bounds. These bounds can be open or closed. All intervals corresponding to the same distance as the distance between these bounds are equally as probable and continuous. If a query is asked a random oracle model will respond to each query uniquely every time that query is submitted. The input of encrypted data could be attacked using machine learning if a pattern of correct guesses is established by the machine learning algorithm. This would imply that the pattern of guessing encrypted data could be easily guessed by a properly trained machine learning algorithm and hence no data could properly be encrypted and hence would be susceptible to cyber attacks.

No proper proof of security for a random oracle model has been established however most black box constructions have been shown to be secure or at least “close enough” but still incredibly secure.[69,72] However, showing that a machine learning algorithm is able to guess encrypted data by one bit with a stronger accuracy than a simple guess or random chance then it is possible to assume that using a stronger machine learning technique or other artificial intelligence to find the patterns in query responses form black box constructions could be used against stronger encryption standards.

If the machine learning algorithm is able to learn the hash function then it is going to be able to predict the output eventually with one hundred percent accuracy. If the function is known then it is possible to calculate the output instantaneously and with one hundred percent probability of getting a correct guess. The machine learning algorithm can continue to ask a query to establish a pattern of correct guesses. If it is able to establish a

pattern of correct guesses then the function can be learned and the problem of predicting outputs becomes easy.

3.3 Snippets of code and Flowchart of Algorithm

Data from training and testing set that has been encrypted with DES is fed into the XGBoost algorithm using Dataframe from pandas library which is a two dimensional data structure. The training and testing data set is a list of integers in tabular form. The training data is fit into a model in XGBClassifier. The predictions on the X test data set are used in a model on the test data set to make predictions which are then evaluated for accuracy. The confusion matrix shows the accuracy results and a decision tree plot is graphed to show the learning algorithm. Using data size N=200 to generate data.

Code with fitted data for XGBoost Algorithm[2] from Jason Brownlee's XGBoost with Python:

```
//begin generate input data  
  
training_data = generate_data(200)  
  
test_data = generate_data(100)  
  
#print(training_data)
```

```

#print(test_data)

X_train = DataFrame([ [ int(x) for x in list(a) ]

for (a,b) in training_data ])

y_train = DataFrame([ int(b) for (a,b) in training_data ])

X_test = DataFrame([ [ int(x) for x in list(a) ] for (a,b) in test_data ])

y_test = DataFrame([ int(b) for (a,b) in test_data ])

print(X_train.shape)

print(y_train.shape)

print(X_test.shape)

print(y_test.shape)

//XGBoost code[2]

# fit model on training data

```

```

model = XGBClassifier()

model.fit(X_train, y_train)

# make predictions for test data
predictions = model.predict(X_test)

# evaluate predictions
accuracy = accuracy_score(y_test, predictions)

print("Accuracy: %.2f%%" % (accuracy * 100.0))

# calculate and show confusion matrix

cm = confusion_matrix(y_test, predictions) #[68,69]
print("Confusion matrix:")

print(cm)

tn, fp, fn, tp = cm.ravel()

```

```
print("TN %d, FP %d, FN %d, TP %d" % (tn, fp, fn, tp))
```

```
# plot single tree
```

```
plot_tree(model)
```

```
pyplot.show()
```

Using the DES Algorithm from:

<https://github.com/RobinDavid/pydes/blob/master/LICENSE.md> [72]. Which is an MIT open source code for DES.

The code for permutations are not shown. To show the key encryption here is main from DES:

```
if __name__ == '__main__':
```

```
    key = "secret_k"
```

```
    text= "Hello wo"
```

```
    d = des()
```

```
r = d.encrypt(key, text)
```

```
r2 = d.decrypt(key, r)
```

```
print(" Ciphared: %r" % r)
```

```
print(" Deciphered: ", r2)
```

Next this Algorithm is needed to convert a string of bits to a string of bytes from pydes[69,72] so the data can be used in the XGboost model.

```
//begin XGBOOST
```

```
import pydes, random
```

```
def string_to_bits(s):
```

```
    return ''.join([bin
```

```
        (ord(i)).
```

```
        lstrip('0b').rjust(8, '0') for i in s]))// [69]
```

```
def bytes_to_bits(bytes):    return
```

```

''.join(format(byte, '08b')

for byte in bytes) //[69]

def generate_data(N):

    d = pydes.des()

    key = "12345678"

    result = []

    for i in range(N):

        indata = random.getrandbits

        (64).to_bytes(8, "big")

        // getting data ready

        indatabits = bytes_to_bits(indata)

        outdata = d.encrypt(key, indata)//encrypted data

```

```

        outdatabits = string_to_bits(outdata)

        result.append((outdatabits, indatabits[0]))

    return result

//Lines of code to show encrypted data being generated

indata = random.getrandbits(64).to_bytes(8, "big")//[72]

indatabits = bytes_to_bits(indata)

outdata = d.encrypt(key, indata)[71]

outdatabits = string_to_bits(outdata)

//Lines of code to shown conversion of string data [69]

import pydes, random

def string_to_bits(s):

```



```

return ''.join([bin(ord(i)).lstrip('0b').rjust(8,'0') for i in s]//[72]

def bytes_to_bits(bytes):

    return ''.join(format(byte, '08b') for byte in bytes)

def generate_data(N):[71,75]

    d = pydes.des()

    key = "12345678"

    result = []

    for i in range(N):

        indata = random.getrandbits(64).to_bytes(8, "big") //[71,72,75]

        indatabits = bytes_to_bits(indata)

        outdata = d.encrypt(key, indata)

```

```
outdatabits = string_to_bits(outdata)

result.append((outdatabits, indatabits[0]))

return result
```

The result is the left DES encrypted data and the indata is the first bit of input. Next, a flowchart of each algorithm and how the inputs are computed through each portion of the algorithm are observed in figure 1 in the following flowchart on the next page, page 51.

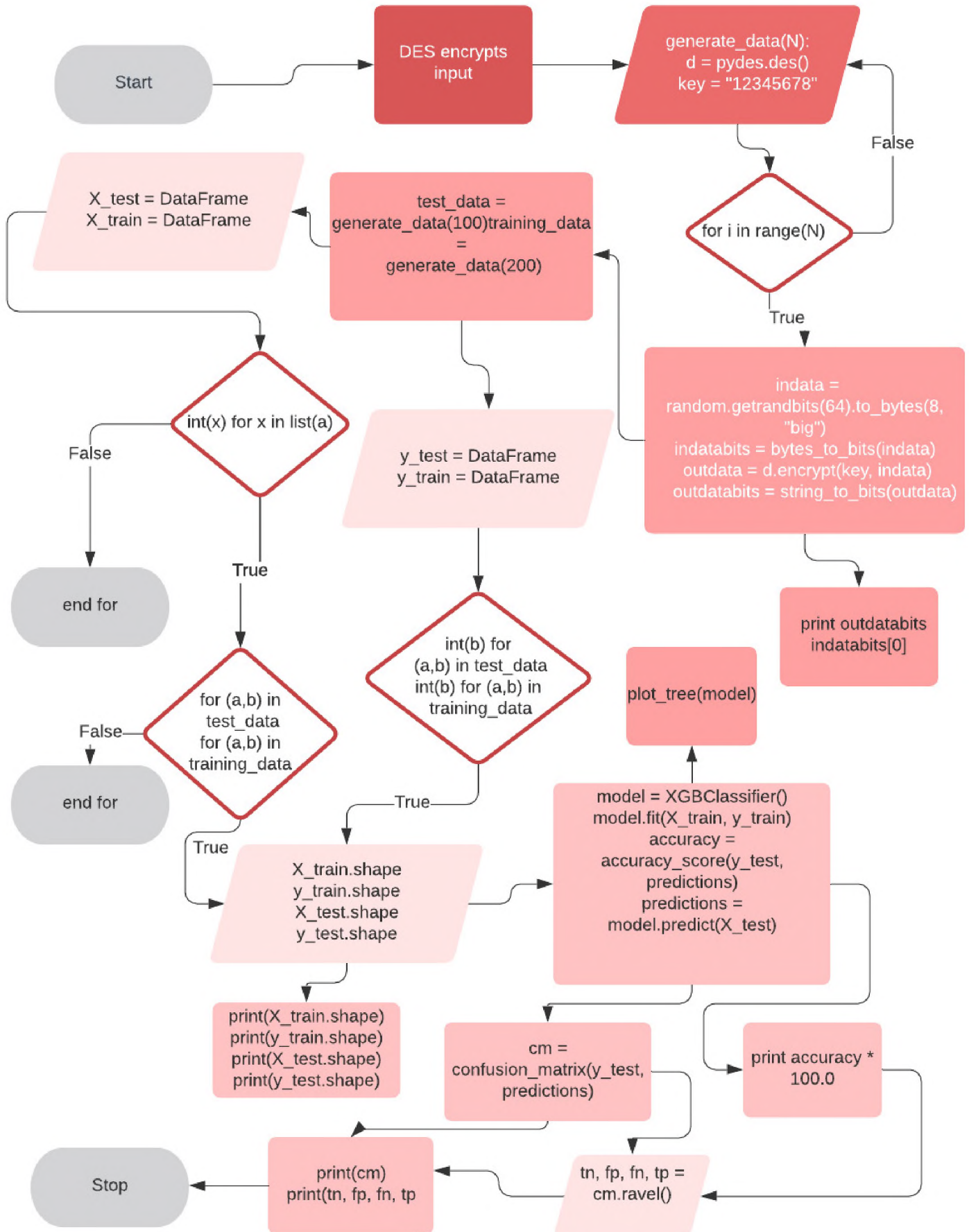


Figure 1: Flowchart of Algorithm with code

3.4 How DES is used to contribute to cybersecurity

DES is a symmetric cipher algorithm which is abbreviated as DES and stands for data encryption standard. It begins as a message block, for example a block of 0's and 1's, and the first 64 bit block is permuated and cut into two pieces as left and right sides. Observe this process pictorially in the following diagram in figure 2. "DES uses a 56-bit key, and maps a 64-bit input block into a 64-bit output block. The key actually looks like a 64-bit quantity, but one bit in each of the 8 octets is used for odd parity on each octet. Therefore, only 7 of the bits in each octet are actually meaningful as a key".[81]

The right side is expanded through an expansion function in the algorithm. 32 bits is converted to a string of 48 bits by a form of permutating that keeps the original order but it is padded with a bit in the front and a bit in the back and there is a pattern to the type of padding. Next the xor function is applied in the algorithm with a sub-key, those 48 bits are expanded and padded bits have a unique function corresponding to a box table which then is shrunk down further to 32 bits, each iteration of permutation corresponds to a different box table."The 64-bit input is subjected to an initial permutation to obtain a 64-bit result(which is just the input with the bits shuffled). The 56-bit key is used to generate sixteen 48-bit per-round keys, by taking a different 48-bit subset of the 56 bits for each of the keys."[81] Another permutation is done and the data is continued to be scrambled.

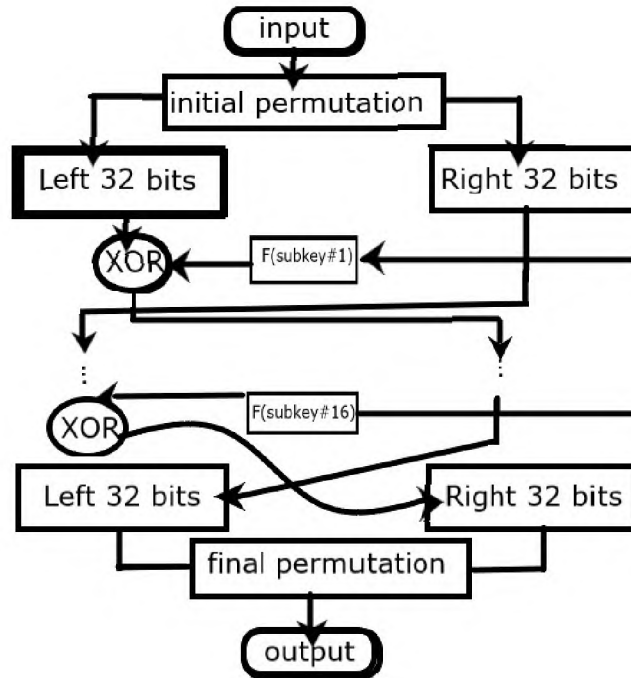


Figure 2: DES Diagram

The goal is to scramble the initial message into some unreadable without a decryption key. "Each round takes as input the 64-bit output of the previous round, and the 48-bit per-round key, and produces a 64-bit output. After the sixteenth round, the 64-bit output has its halves swapped and is then subjected to another permutation, which happens to be the inverse of the initial permutation." [81] The table data is once again applied with the xor function with the left side. The right side has a table that is outputted then again the xor function is applied with the left side which is then put into the new right side then the right side from the previous round and put it in the left side that's it do it 15 more times total of 16 rounds. At end reverse left and right then do another permutation and encryption is done a ciphertext is finally obtained.

What if a type of cryptanalysis could be done that could guess the pattern of permutations done or the probability of an output from each stage of permutation? The

contribution of this research is to show that it is probable with an accuracy of above a random guess to use a machine learning algorithm to decrypt at least the first bit of data from the final output of a DES encrypted message. If the number of permutations is known then further research could be used to conclude that the output of any stage of the permutation cycle could be decrypted using machine learning or even used on a stronger encryption function. Permutations are very common in the study of cryptography so using a machine learning technique to calculate the probability of each output from each stage of permutations could prove useful in cryptography and further research in cryptanalysis. The permutations produce a cipher from a transposition and the letters are not changed in plaintext. The shift cipher substitutes and changes the letters in plaintext. It could be possible using the result of this thesis to conclude that an artificial intelligence technique, in particular a machine learning technique, could be used to deduce the probability of these possible substitutions.

3.5 How Machine Learning is used to contribute to cybersecurity

There are advanced machine learning techniques used to contribute to cybersecurity. Machine learning algorithms are used to advance techniques for decryption in the field of cybersecurity and will contribute significantly to the quick paced research needed in this field. The question becomes how to select a good machine learning technique that can contribute to the advancement of the field of cybersecurity. There are several machine learning algorithms available, and the choice is unique to the specific task that is being solved and also it is unique to the type of problem that is being solved. Writing an algorithm to crack

encryption codes requires a good choice in the machine learning technique being used.

There were many selective choices for which machine learning technique to use. How to train to train the neural network or which neural network is best for the task is a problem that needs to be considered. For this experiment a labeled data set is needed to use it as training data set and test data is needed as a validation data set for tuning hyper parameters where activation functions weight data to process and pre-process normalization optimization functions. KNN won't work because integer distance lazy training not needed, CNN is also not an acceptable choice since it is used for image recognition and the input data for this experiment is tabular data. When training a neural network two fundamental things are required: a feed forward algorithm and a back-propagation algorithm. A feed forward algorithm calculates the output vector using the input vector. A back-propagation algorithm trains the weights of the neural network to output an adjusted weighted vector. The best choice for a machine learning algorithm for this experiment is a supervised learning technique that is specifically perfect for tabular data. Classification problems use neural networks to classify observations into two or more classes. To successfully implement a machine learning strategy on a classification problem the best choice is to use a weak learner performs slightly better than random chance. Gradient boosting is a supervised learning algorithm for classification problems that uses an ensemble of decision trees that are weak learners in sequence until the most accurate model is possible. Gradient boosting is a powerful machine learning algorithm and XGBoost is an extreme implementation of that. One ensemble is a single learner in the sequence and the supervised learning algorithm is trained to make predictions on unseen data. Each step of the sequence involves putting

more weight on difficult to classify patterns and higher focus is put on training more difficult patterns. It is numerically optimized by minimizing the loss function.

3.6 Better than a coin toss

It could be possible for a robot to learn entirely on its own without human programming since machine learning techniques and artificial intelligence would hopefully improve on a machine understanding to develop its own learning from interacting with its environment and should not rely on human intervention. "It may help decide where to run code, where to keep state, when to start up a new execution environment, and how to keep utilization high and costs low while meeting performance objectives." [79] It should be able to learn and formulate independent ideas. Machines may even one day learn human social behaviors and could learn how to lie to humans or act human. This type of ideology should not be limited to just the social realm but could extend to encryption techniques or decryption strategies. The future of cloud computing depends on strong encryption techniques that can withstand stochastic cyber attacks. "This path may prove to be an important advantage because it allows developers to reason about security at a higher abstraction level." [79] Using the results of this research from this thesis it has already been shown that a machine learning algorithm can be applied to predict encrypted data with an accuracy percentage that is significantly higher than a random guess or the probability of guessing heads out of tails in a toss of a fair coin. It could be used to help build an architecture that could support these types of cyber attacks on a system and help train other models. "Machine learning can help optimize serverful computing too, but serverless abstractions give cloud providers more

control over the relevant knobs, as well as the visibility across many customers required to train robust and effective models.”[79] Robots would have to be trained to prevent malicious attacks from predicting encrypted data. Key based encryption may become outdated since it depends on each client or employee to have their own key into the system. With serverless computing it is necessary to prevent against malicious hacking attempts. ”They do not need to implement lower-level security mechanisms, which could lead to fewer security mistakes. While this benefit must be weighed against the exposure to attacks through shared hardware, we believe that improved abstractions may eventually make application security easier to achieve with serverless computing.”[79] If robots are able to predict inputs then machine learning algorithms can be used to predict against attacks from other machine learning tactics that may be malicious. Since machine learning can be applied in a security platform in serverless computing. Machine learning algorithms can be used to predict and prevent vulnerabilities in a system. It is important to see the flow of the XGBoost algorithm because it is a critical aspect of this application of machine learning. This choice of algorithm was a good choice because of the labeled data set that was encrypted by the DES algorithm. Note the stepwise algorithm in the following table.

Algorithm 1 XGBoost Algorithm

XGBoost

Import libraries

Prepare data

Load the datasets

Fit model on training data

Fit Data to model

Run the fitted model

make predictions for test data

evaluate predictions

calculate and show confusion matrix

plot single tree

3.7 Description in detail of contributing to security

Decryption can be used against ransomware attacks. End to end encryption is no longer a secure encryption method. Cloud security is a reasonable solution. What would it mean if an artificial intelligence agency would be able to cross into the path of human social interaction? It could mean a reasonable destruction of end to end encryption or at least a reasonable argument to lobby for different encryption standards. Since if a machine learning algorithm is successfully able to predict the input of data that has been encrypted by a symmetric encryption it is reasonable to assume that further researcher practices would lead to more predictive results against other machine learning tactics to predict inputs. Hence, using end

to end encryption may become prehistoric. This in particular is of great significance.

3.8 Cloud Encryption and Cloud Security

What would it mean if artificial intelligence intersected with human communication. Would it be more secure or less secure? It could mean a reasonable destruction of end to end encryption or at least a reasonable argument to lobby for different encryption standards. Since if a machine learning algorithm is successfully able to predict the input of data that has been encrypted by a symmetric encryption it is reasonable to assume that further researcher practices would lead to more predictive results against other machine learning tactics to predict inputs. Hence, using end to end encryption may become prehistoric especially with the introduction of serverless computing.”Today, several serverless environments can run arbitrary code, each catering to a particular use case.”[79] Cloud computing allows access for a broader net of communication so it should be protected for each use case. If there is a vulnerability for one user then the entire cloud is at risk for cyber attacks.”An attacker observing the size and timing of network traffic, even if it is encrypted, might make inferences about private data. Addressing these risks may be possible through oblivious computing.”[79]

Since if a machine learning algorithm is successfully able to predict the input of data then it could mean a reasonable destruction of end to end encryption or at least a reasonable argument to lobby for different encryption schemes for cloud computing or severless architecture. ”The future evolution of serverless computing, and in our view of cloud computing, will be guided by efforts to provide abstractions that simplify cloud programming.”[79]

The description of contributing to security includes the benefits of being able to use machine learning in decryption strategies since decryption can be used against ransomware attacks.”In serverless computing, programmers create applications using highlevel abstractions offered by the cloud provider.”[79] In cloud computing the applications are created using highlevel abstractions and are not encrypted on the programmers device. ”Google Cloud Dataflow and AWS Glue allow programmers to execute arbitrary code as a stage in a data processing pipeline, while Google App Engine can be thought of as a serverless environment for building Web applications.”[79] Encryption schemes that protect a cloud environment may be unsecure if they are limited with end-to-end encryption especially if machine learning techniques can be used to predict the input without knowing a secret private key.

Cloud computing allows security concerns to be updated without client involvement. Cloud computing or serverless computing that take on some of the security responsibilities and increase security of a system. ”Serverless computing merely shifts some security responsibilities from the cloud customer to the cloud provider, just as it shifts other system administration responsibilities. With cloud functions, security updates to operating systems, language runtimes, and standard software packages are applied without customer involvement, usually quickly and reliably.”[79] If each serverless environment is protected by a single encryption key it may not be immune from cyber attacks especially a cyber stochastic attack. ”Serverless computing leads to fine-grained resource sharing and so increases the exposure to sidechannel attacks, whereby attackers exploit subtle behaviors of real hardware that differ from either specifications or programmer assumptions.”[79]When the system is better at responding to changes in the system than its administrators means that the system

can be vulnerable to cyber attacks. A machine learning algorithm could override security responsibilities. "It may also aid in identifying malicious activity that threatens security, or in automatically cutting up large programs into pieces that can execute in separate cloud functions." [79]

4 Experiments and Justifications

4.1 Stochastic Cyber Attack

A stochastic attack by a machine learning algorithm is the behavior of many machine learning algorithms to predict with some accuracy above random chance of guessing the input of some encrypted data set. The random probability that a machine learning algorithm could accurately guess the input of an encrypted data is an example of a stochastic attack using a machine learning algorithm. Machine learning algorithms typically behave randomly as a variable process where the outcome involves a degree of random chance or general randomness. This property of randomness is called stochastic in probability and statistics which is a random process or event where something random is happening to a mathematical object or variable.

The purpose of this experiment has been shown to examine how machine learning algorithms will reverse engineer hash functions which are used to encryption and are assumed to be impossible to reverse. The DES algorithm is directly imported from MIT's opensource library. The XGBoost algorithm is from Jason Brown's Machine Learning Mastery blog and ebook.

4.2 DES weak encryption standard

DES is a symmetric encryption algorithm that is considered a weak one way function.

The algorithm for implementation is:

1. Pydes from mit library permute for data permute on key permutation [71,75]
2. Generate pydes [71] string to birs data encrypted in data with key , out data
3. Next part training testing keep t as last bit of input split training x,y and testing x,y
4. Do pydes encrypt with key
5. T in data bits outdates bits
6. T' outdatabits, indatabits
7. Training and test sets known
8. Data ready for machine learning algorithm

4.3 Using ML to break DES Encryption

In ML a common task is the study and construction of algorithms that can learn from and make predictions on data, and these algorithms function by making data driven predictions or uses greedy decision-based ensemble techniques through building a mathematical model from input data. "Breaking DES would mean finding a key that maps that plaintext to that ciphertext." [81]

1. The first task is to convert 64 bits to bytes
2. T' is outdata (des encrypted) indata[0] (first bit of input)
3. T' is fed in xg [73]
4. Each time you run it you get different xo
5. The machine learning algorithm (XGBoost[73]) will be run each time for testing against training
6. Then confusion matrix measures the last run exp
7. The training data set is the term for the samples used to create the model.
8. The test set is used to qualify performance.

4.4 Running the experiment

Experiment Steps through the algorithm:

1. Basically, just imported XGBoost fed in T' model is fit on training data then predictions made on test evaluate predictions and calculate confusion matrix.
2. XGBoost uses advanced regularization which improves model generalization training is fast and can be parallized/distributred across clusters.
3. Regularization is the process of adding information to prevent overfitting. This is done in the xgBoost library[2].
4. Load file as numpy array using the numpy function loadtxt
5. Input patterns x outputs patterns y
6. Split the x and y data into training and test data set
7. Training set is used to prepare XGBoost model and the test set will be used to make new predictions from there we can evaluate the performance of the model.
8. Use train-test-split see gives same split of data next train the model.
9. Splitting original datasets into input and output columns and call them x and y respectively.

10. Next call the function passing both arrays and split them into train and test data subsets. Shape of data is a tuple. “In mathematics, a tuple is a finite ordered list (sequence) of elements. An n-tuple is a sequence (or ordered list) of n elements, where n is a non-negative integer.” [18]
11. Models are fit using the skikit learn api and the model.fit function parameters for training the model can be passed to the model can be passed to the model in the constructor.
12. Use the trained model to make predictions then after we have used the fit model to make predictions by comparing them to the expected values -using accuracy function
13. Confusion matrix is calculated. [68]
14. Accuracy scores are printed
15. Decision tree is plotted

4.5 Algorithm Description and explaining the experiment

First, DES[72] is used to encrypt data of fixed size. Original message broken in 64 bit chunks. Permutation occurs over split data, left split to xor function then xor applied to right and right breaks to left side and is expanded xor with a subkey to right side of 48 bits which shrinks from splitting and permutation occurs again then xor applied to right side. Repeated 16 times for each subkey. Right and left sides are split from permutation and reverse ciphertext. Two parts to the encryption: Left creation key and Right block cipher. Key converts to binary, and data type converted from Ascii to hex to bits. DES[72] encrypts a bunch of 64 bits and Uses key of 56 bits. First input is the first 64 bit value. The key is a string key 8 bytes long. To prepare data for the machine learning algorithm, print out the 0 and 1's and generate a random string. Everytime you run it, it is random in that the output will not be dependent on the previous run.

Each implementation produces a different and random output. The test data is not part of the training data. The neural network acts as a "recognizer" to a list of tuples of two strings. First string is encrypted and the second is clear text data from which it is encrypted. Data is split into two tables. The test side ignores the input, the model is fed the strings. The pydes.py is the DES[72] algorithm encrypts the training data, N=200 or N=100 or N=300 for each test case, where N is a fixed data size, and the test side ignores the input and is 100 elements to make the output.

Training:

```
('001001100101010001000111110011001111101110011000000000000101100', '0'),  
...,  
('0101000010000111001100000111100011000111101010101101011111000011', '0')
```

Testing:

```
('1011001010011101110110010100001001011011111101010101101100010000', '0')  
...,  
('0000100110010100000101011010010111010001111001000001100010100011', '1')
```

The python languages recognizes this as a list and it is passed into the program and transferred from generate.py which generates the encryption data. Lambda is a string break it down into a list. Data is split again, X is a list of 64 bits (0,1) Y is just (0,1) for 100, 200, or 300 rows training data 100 rows test data as a string of string to list of list data type. Need trainable neural network that reads a string of bits, so it is necessary to use Panda dataframe from list to convert data to list to numpy using Dataframe which applies change to "N". Output do.py which is the conversion algorithm and used to train in the XGBoost[2] algorithm. Do.py runs again randomly but they have same encrypt and same algorithm. Do.py can be used to copy and paste into training data but for efficiency we need a better way to fit data into algorithm other than copy paste, so that is why the code for data converting is created and used in the XGBoost[2] algorithm. Do.py gives 100 more rows then you have 200 more More training data can be used to improve accuracy and can be repeated over and over.

Using XGBoost[2] as the machine learning algorithm to fit model on training data, then make predictions for test data, and evaluate those predictions and the accuracy of those predictions. Accuracy percent score comes from its training data set. Explaining the table of data: 0101010101 is on the left which is des encrypted and '1' on the right is the first bit of input for N=300 or 200 or 100 depending on datasize. Data is generated. Accuracy should skip for arbitrary N since N is a fixed data size, using 200 for example, data size can change, and size 100 for test data , creating the confusion matrix from correct guesses of 0,1. No real imbalance, random each time true 1.

Calculate confusion matrix and for observation print result. Example in accuracy calculation: $20 + 33 = 53$ is the accuracy and $TN + TP = 11 + 36$ is the balance. Then plot a tree using graphviz which is a library imported, only if encryption is good from pydes.py[72]

Conclusion: Using Machine learning algorithms proves DES is vulnerable to stochastic attack

4.6 pydes.py

Start with pydes.py [71] which comes from mit library. The entire code was not used, only the part of the algorithm that includes the des algorithm and necessary permutations to encrypt the required input. Using the des algorithm from the opensource mit library, a table of 0's and 1's was encrypted. The program pydes uses an algorithm that generates all the keys, applies initial permutation on the keys, splits it in to left and right and apply 16 rounds of permutation. Apply the shift associated with each round and merge them. Apply another permutation to the key to get the last key. Add padding to the data and remove padding of

the plain text. The des algorithm now encrypts an input.

4.7 Do.py

The des algorithm now encrypts an input. Next, there needs to be a program to import des in python and use it to encrypt a string of bits and return a string of bytes.

4.8 Generate.py, nextpart.py

Using a simple script to generate the data to be encrypted the pydes algorithm is imported and used with some set key. The indata comes from the random library, and 64 bits to bytes becomes the indata. The outdata is the encrypted data with the key and in data and is a string. The result is an appended table of outdata bits and the first bit of the indata. The next part is we set our training and test data which is our table from the encrypted des algorithm. Using dataframe from pandas library to convert the data into something readable that can be plugged into the machine learning algorithm Once the data is split into training and testing the xgBoost algorithm is applied [73].

4.9 xg.py

The XGBoost algorithm[73] is used for fitting the model on the training data to fit x train and y train and then making predictions for test data, where the predictions are done on the x test set. The model is than evaluated to make predictions and the accuracy score measures

the y test set. To interpret the results a confusion matrix[68] is used from the y test set and predictions to show the true negatives, false positives, false negatives, and true positives respectively. The code for the tree plot is also included in the XGBoost algorithm. [73]The plot of a single tree is also shown to provide further insight on the experiment.

4.10 Value of using XGBoost

The XGBoost algorithm is a gradient boosting decision tree machine learning algorithm that uses greedy optimizing techniques for predicting the value of a model throughout steps in the learning process. The trees in XGBoost are built one after the other in a sequence such that each tree corrects the errors of the previous tree and the last tree corrects the final error. As the tree grows each tree next in the sequence learns from the residuals in the algorithm. XGBoost is a favorite algorithm among many people in the machine learning industry. “Efficient algorithm for converting relatively poor hypothesis into a very good hypothesis”. [2] A weak learner has a performance that is slightly higher than random chance. It weights each instance by difficulty and performance, when new weak learners are added in the sequence they perform in a greedy way and focus training on the most difficult. Boosting works by optimizing the loss of the stagewise additive models by adding weak learners one at a time and the learners already in are left unchanged. The loss function is optimized, the weak learner makes predictions, and the stagewise additive model continues to minimize the loss function by adding weak learners, decision trees are used such that they output values at the split and add them together in the next predictive model.

Each output adds the corrected residuals in each predictive output. The trees are added one at a time and gradient descent is used to minimize a set of parameters which minimizes the loss when adding trees. The parameters in this model are the weak learners or decision trees. After each error is calculated the weights are updated. Each stage of the calculation minimizes the error and each tree is added to reduce the loss. The weak learners are modified in right right direction by residual loss and produced a weighted classification. XGBoost uses gradient boosting decision tree algorithms by automatic handling missing data and block structure that supports the parallelization of the tree algorithm construction that is trained and boosted very quickly to fit the model on each new stage of data. “Dominates structured or tabular datasets on classification and regression predictive modeling problems.”[2]

4.11 DES encryption standard as weak algorithm

DES has been proven to be a weak encryption algorithm. Using a weak encryption algorithm for the purpose of this project can hopefully provide further insight into how machine learning algorithms can be used to decrypt stronger encryption methods.

4.12 Guessing first bit

Everytime the algorithm is implemented the results are randomly generated. If the implementation can produce an accuracy score with probability .01 percent above 50 percent at least once then the experiment is a success. Several implementations were run on the command prompt and the accuracy scores were printed in the command prompt and a screenshot was taken of four results to demonstrate the success of the experiment. Also shown are several plots of the decision tree which are weak learners in the machine learning algorithm and were used in each implementation respectively. The first bit of the input that was encrypted with the DES algorithm has been predicted with over 50 percent accuracy as shown in figure 6, 8, 10, 12, 14. Which means the machine learning algorithm, XGBoost, is able to successfully guess the first bit of encrypted data with accuracy scores better than random chance or a pure guess.

4.13 Confusion matrix from three test runs

Accuracy scores give the true negatives added with the true positives. The confusion matrix shows the true positives and true negatives as shown in the accuracy score but also gives the residuals of false positives and false negatives. A confusion matrix is useful since it allows human researchers to understand the results.

The experiment will be random each time it runs. Here is a sample of the results from three concurrent test runs for $N=200$, input data = 64 bits, and output data = 1 bit.

TN means true negative, FP means false positive, FN means false negative, TP means true positive.

1) Accuracy 45.00 percent with confusion matrix:

[[25 25]

[30 20]]

TN 25, FP 25, FN 30, TP 20

Here the accuracy score is 45 percent. The true negatives are 25 and true positives are 20. When added together that gives a score of 45 percent. The false positive is 25 and false negatives is 30. which adds to a total of 100 percent.

2) Accuracy 52.00 percent with confusion matrix:

[[31 27]

[21 21]]

TN 31, FP 27, FN 21, TP 21

Here the accuracy score is 52 percent. The true negatives are 31 and true positives are 21. When added together that gives a score of 52 percent. The false positive is 27 and false negatives is 21. which make up the residual.

3) Accuracy 53.00 percent with confusion matrix:

[[31 19]

[28 22]]

TN 31, FP 19, FN 28, TP 22

Here the accuracy score is 53 percent. The true negatives are 31 and true positives are 22. When added together that gives a score of 53 percent. The false positive is 19 and false negatives is 28 which make up the residual.

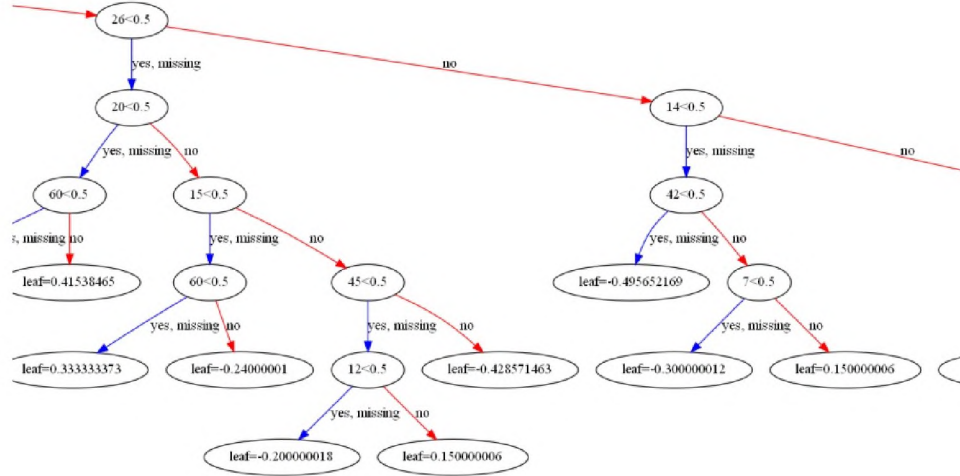


Figure 3: close up of decision tree nodes and branches

4.14 Decision tree plot

Here in figure 3 is an example of a decision tree from an the accuracy score of 55 percent. The true negatives are 25 and true positives are 30. When added together that gives a score of 55 percent. The false positive is 26 and false negatives is 19. which make up the residual which adds to a total of 100 percent. Below are the figures for the decision trees in this run of the experiment.

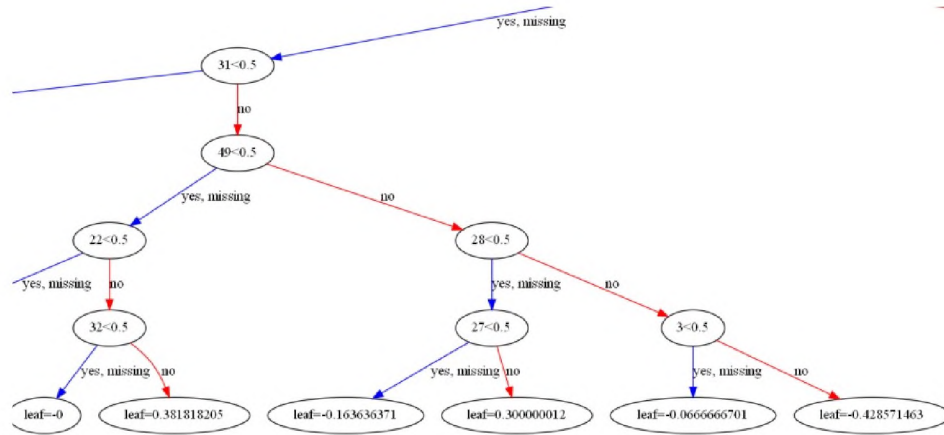


Figure 4: close up of decision tree nodes and branches

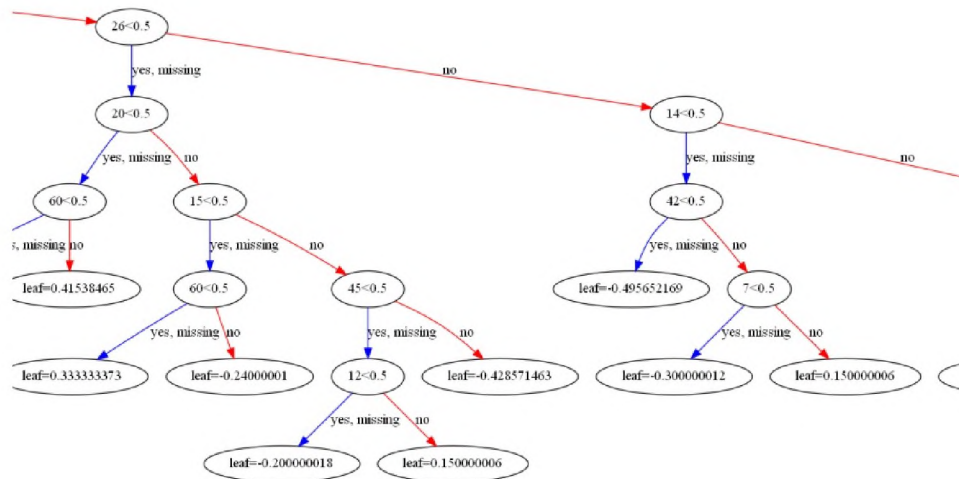


Figure 5: close up of decision tree nodes and branches

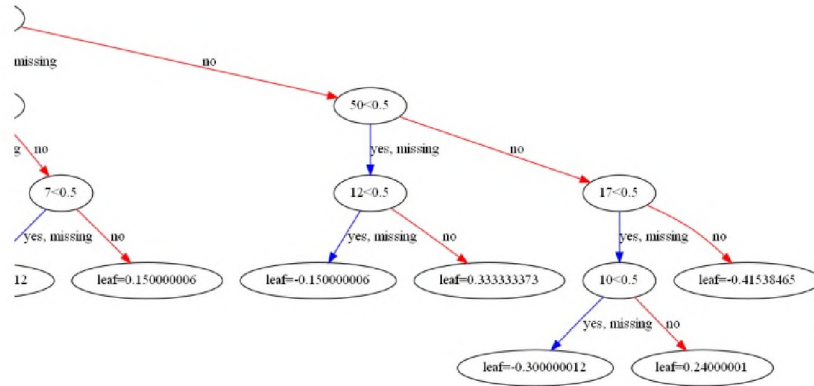


Figure 6: close up of decision tree nodes and branches

Next are the decision trees for an experiment where the accuracy score is 61 percent. The true negatives are 27 and true positives are 34. When added together that gives a score of 61 percent. The false positive is 18 and false negatives is 30. which make up the residual which adds to a total of 100 percent.

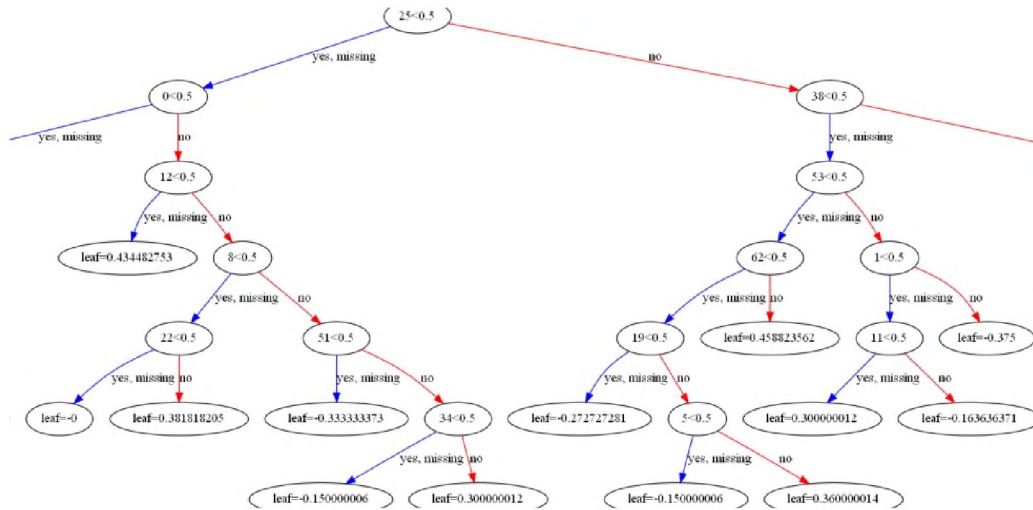


Figure 7: Decision tree corresponding to Figure 15

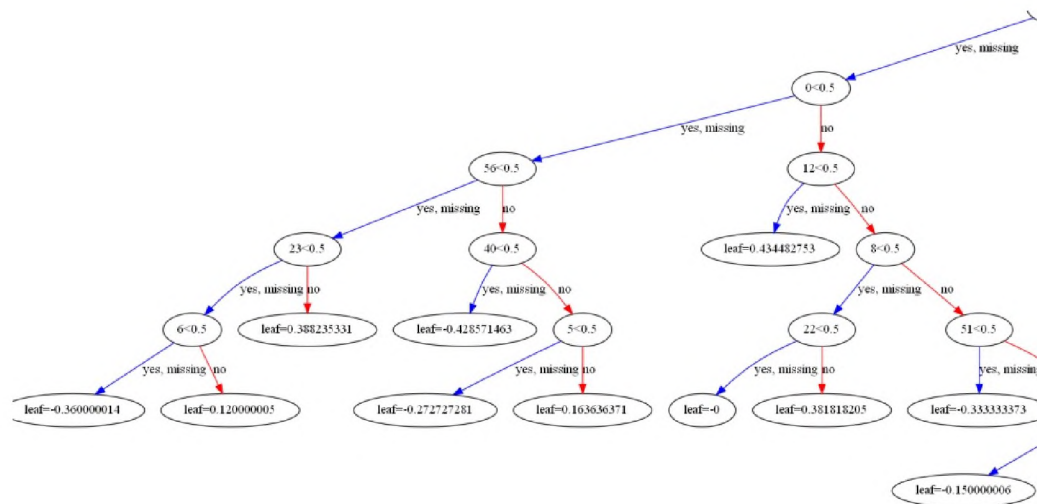


Figure 8: close up of decision tree nodes and branches

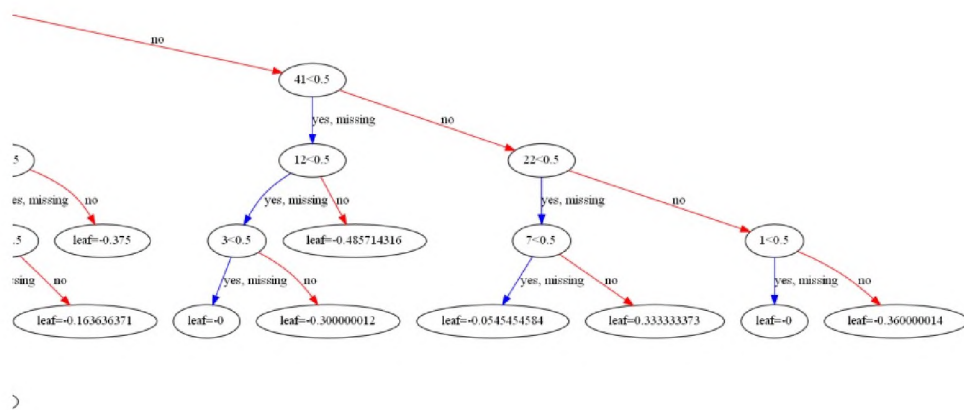


Figure 9: close up of decision tree nodes and branches

5 Conclusion and future work

The contributions of this thesis are extremely valuable. This experiment uses machine learning to predict the first bit of encrypted data and has shown that it can be done with a higher accuracy than 50 percent which means it can be predicted better than guessing. Using a weak encryption standard such as DES can give us insight into how machine learning can help break encryption standards. Further research could conclude that machine learning or other artificial intelligence agencies can be used to perform malicious attacks against a system with stronger encryption standards leaving a system unequipped against stochastic attacks.

The conclusion to this thesis is to state that it is possible to use machine learning to predict the input of data with a weak encryption algorithm such as DES. It has hence been shown that it is possible, using machine learning, to reverse engineer a hash function which should be impossible to reverse. Use machine learning to predict the first bit of encrypted data can be done with a higher accuracy than 50 percent which means it can be predicted better than guessing. Finding new ways to provide security is important to the future of cloud security. New methods in encryption need to evolve as technology evolves otherwise the worlds data is at risk for exposure.

Machine learning can help with reverse engineering hash functions which are assumed to be irreversible one-way functions. This experiment has shown that we can do better than 50 percent at guessing the inverse which means a machine learning algorithm

can do better than a guess at reversing the first bit of an encrypted message. DES is a weak encryption function, but this experiment has given insight into the potential to try a more advanced encryption function such as switching DES with AES a better, stronger encryption standard. This same line of research can be conducted with a standard symmetric encryption function. With more than 50 percent accuracy we can do better than guessing the inverse if we tried this same line of research on a more advanced function such as AES256. A question arises: What could be the accuracy of predicting two bits of encrypted data? Or more? Or even if it is possible for a machine learning algorithm to predict accurately an entire input. Or another question: What could be the accuracy of predicting one bit of encrypted data that has been encrypted with a stronger encryption method such as elliptic curve encryption or a stronger symmetric encryption standard such as AES256.

The future works for this line of research could be to recommend a stronger encryption method for cyber security professionals such as elliptic curve cryptography, especially in wireless communication. Furthermore, applying machine learning to stronger encryption standards such as AES-256 or SHA-256 would provide useful insight to how powerful our current encryption standards are when faced against artificial intelligence or stochastic attacks. A program to calculate the average rate of accuracy at guessing the first bit could provide useful insight into the probability of a machine learning algorithms aptitude to consistently make the correct guess. It could also be useful to investigate the outliers and explain the probability of a guess of over 60 percent or under 30 percent . Although the guesses on average appear to be around 50 percent, using a more advanced stochastic measurement of the average accuracy of guessing could provide useful insight.

Government agencies rely on strong encryption methods to protect against cyber-attacks. The AWS cloud environment uses AES-256 [35] and SHA-256 as their encryption standard and XGBoost is possible to run on an EC2 instance. An enticing sequel to this line of research could be to provide empirical statistics for the accuracy of guessing the first bit of encrypted data especially using the XGBoost machine learning algorithm in the AWS cloud environment.

References

- [1] Michael Soltys. *Cybersecurity in the AWS Cloud*. CoRR vol abs/2003.12905, 2020
<https://arxiv.org/abs/2003.12905>
- [2] Jason Brownlee. *XGBoost With Python Gradient Boosted Trees With XGBoost and scikit-learn*. 2021
- [3] CM Schneider et al. *Mitigation of malicious attacks on networks*. P. Natl. Acad. March 8, 2011 vol. 108. no. 10
- [4] J. Harris. *Algebraic Geometry: a first course*. Springer, GTM 133, 1992.
- [5] G. Salmon. *A treatise on the analytic geometry of three dimensions*. Dublin: Hodges, Smith, Co., 1865.
- [6] Grzegorzcyk: Elliptic Curve
https://csuci.blackboard.com/bbcswebdav/courses/2172_MATH_584_12576/elliptic_curve_addition.ppt
- [7] Garrity Thomas. *Algebraic Geometry: A Problem Solving Approach*. American Mathematical Society 2013
- [8] graphing utility
[desmos.com](https://www.desmos.com)
- [9] [graui.de/code/elliptic2](https://www.graui.de/code/elliptic2)
- [10] [wolframalpha.com](https://www.wolframalpha.com)

- [11] <https://www.quora.com/What-is-a-tangent-space>
- [12] www.math.tamu.edu > boas > courses > solution2
- [13] Mahmoody, Mohammad, Tal Moran, and Salil Vadhan. *Publicly Verifiable Proofs of Sequential Work*. In Proceedings of the 4th Conference on Innovations in Theoretical Computer Science, Berkeley, CA, January 9-12, 2012.
- [14] Michael Soltys *Feasible proofs of Szpilrajn's Theorem A proof-complexity framework for concurrent automata*. Journal of Automata, Languages and Combinatorics, 16(1):27-38, 2011.
- [15] Rafael Pass. *Lecture 21: Collision-Resistant Hash functions and General Digital Signature Scheme*.
- [16] Wikipedia.org
- [17] <https://searchsecurity.techtarget.com/definition/cipher>
- [18] <https://searchsqlserver.techtarget.com/definition/hashing>
- [19] <https://gcn.com/articles/2013/12/02/hashing-vs-encryption.aspx>
- [20] <https://www.scientificamerican.com/article/the-mathematics-of-hacking-passwords/>
- [21] Jung: Hash

<https://www.cs.usfca.edu/~ejung/courses/686/lectures/05hash.pdf>
- [22] Jong-In Lim Dong-Hoon Lee. *Information Security and Cryptology – ICISC*. 2003 6th International Conference, Seoul, Korea, November 27-28, 2003. Revised Papers

- [23] Jason Brownlee. *Deep Learning with Python*. 2020
- [24] Jason Brownlee. *Introduction to Time Series Forecasting with Python* . 2021
- [25] Jason Brownlee. *Machine Learning Mastery With Python*. 2021
- [26] Jason Brownlee. *Data Preparation for Machine Learning*. 2021
- [27] Jason Brownlee. *Ensemble Learning Algorithms With Python*. 2021
- [28] Jason Brownlee. *Imbalanced Classification with Python*. 2021
- [29] Jason Brownlee. *Machine Learning Algorithms From Scratch*. 2021
- [30] <https://towardsdatascience.com/regularization-an-important-concept-in-machine-learning/>
- [31] <https://www.netsuite.com/portal/resource/articles/financial-management/predictive-modeling/>
- [32] Hirose S. *Yet Another Definition of Weak Collision Resistance and Its Analysis*. In: Lim J.L., Lee D.H. (eds) *Information Security and Cryptology - ICISC 2003*. ICISC 2003. Lecture Notes in Computer Science, vol 2971. Springer, Berlin, Heidelberg. 2004
- [33] AWS dashboard:
<https://awsacademy.instructure.com/>
- [34] <https://www.moviequotes.com/s-movie/i-robot/>
- [35] <https://cathyatseneca.gitbooks.io/data-structures-and-algorithms/content/tables/hashtable.html>
- [36] <https://www.solver.com/regression-tree>

- [37] <https://towardsdatascience.com/boosting-algorithm-xgboost-4d9ec0207d>
- [38] <https://searchsecurity.techtarget.com/definition/cybersecurity>
- [39] <https://www.cryptomathic.com/news-events/blog/symmetric-key-encryption-why-where-an>
- [40] <https://digitalguardian.com/blog/what-data-encryption>
- [41] Science Direct: Symmetric Key Encryption
<https://www.sciencedirect.com/topics/computer-science/symmetric-key-encryption>
- [42] <https://blog.mailfence.com/symmetric-vs-asymmetric-encryption/>
- [43] <https://www.vtscada.com/help/Content/Scripting/Tasks/proEncryptionAndDecryption.htm>
- [44] <https://dataoverhaulers.com/can-encrypted-data-be-hacked/>
- [45] <https://medium.com/searchencrypt/what-is-encryption-how-does-it-work>
- [46] <https://www.open.edu/openlearn/science-maths-technology/computing-and-ict/systems-co>
- [47] <https://sectigostore.com/blog/5-differences-between-symmetric-vs-asymmetric-encrypt>
- [48] <https://www.tutorialspoint.com/difference-between-private-key-and-public-key>
- [49] Mohammed Nazeh Abdul Wahid, Abdulrahman Ali, Babak Esparham and Mohamed Marwan. *A Comparison of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blow-fish for Guessing Attacks Prevention*. August 10, 2018
- [50] Thomas Icart. *How to Hash into Elliptic Curves?*. 2009
- [51] <https://www.thesslstore.com/blog/what-is-256-bit-encryption/>

- [52] https://www.youtube.com/watch?v=S9JGmA5_uY
- [53] <https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard>
- [54] <https://www.eetimes.com/how-secure-is-aes-against-brute-force-attacks/>
- [55] <https://www.cryptologie.net/article/389/a-hash-function-does-not-provide-integrity/>
- [56] Science Direct: Hash Collisions
<https://www.sciencedirect.com/topics/computer-science/hash-collision>
- [57] Confusion Matrix Machine Learning:
<https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/>
- [58] Paul Keeler. *Notes on stochastic processes*. March 20, 2018
- [59] Bram Cohen. *AES-hash*. May 2, 2001
- [60] Google Support
<https://support.google.com/google-ads/answer/9004655?hl=en>
- [61] Preneel B. *Second preimage resistance*. Van Tilborg H.C.A. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA . 2005
- [62] <https://youtu.be/3jGMCyOXOV8>
- [63] <https://youtu.be/fEKdpsCbtC8>
- [64] <https://youtu.be/Sy0sXa73PZA>

[65] Javatpoint: Differences between AI and Machine Learning

<https://www.javatpoint.com/difference-between-artificial-intelligence-and-machine-learning>

[66] Julianna Delua. *Supervised vs. Unsupervised Learning: What's the Difference?*. SME,

IBM Analytics, Data Science/Machine Learning March 2021

[67] Supervised Vs Unsupervised Learning

<https://www.guru99.com/supervised-vs-unsupervised-learning.html>

[68] Sklearn: Confusion Matrix

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

[69] Stackoverflow: String to list to Bits

<https://stackoverflow.com/questions/10237926/convert-string-to-list-of-bits-and-vice-versa>

[70] Python DES Project

<https://pypi.org/project/des/description>

[71] Robin David GITHUB python DES

<https://github.com/RobinDavid/pydes/blob/master/LICENSE.md>

[72] Stack Exchange: Hash Function on Finite Fields

<https://math.stackexchange.com/questions/1347240/how-to-attack-universal-hash-function>

[73] Jason Browne: Machine Learning Mastery XGBoost

<https://machinelearningmastery.com/xgboost-with-python/>

[74] Pandas Python Data

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.hist.html>

[75] GITHUB CODE RESOURCE

<https://github.com/kongfy/DES/blob/master/Riv85.txt>

[76] Martignon: Security Systems DES

<https://www.lri.fr/~fmartignon/documenti/systemesecurite/4-DES.pdf>

[77] IBM: AI VS MACHINE LEARNING

<https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-netw>

[78] Seyedsaman Emami, Gonzalo Martínez-Muñoz *Sequential Training of Neural Networks*

with Gradient Boosting. February 2020

[79] Johann Schleier-Smith, Vikram Sreekanti, Anurag Khandelwal, Joao Carreira, Neeraja

J. Yadwadkar, Raluca Popa, Joseph Gonzalez, Ion Stoix, David Patterson.

The evolution that serverless computing represents, the economic forces that shape it, why it could fail, and how it might fulfill its potential.. 2021

[80] Vanessa Sochat. *The 10 Best Practices for Remote Software Engineering*. 2021

[81] Mike Speciner, Radia Perlman, Charlie Kaufman. *Network Security: Private Commu-*

nications in a Public World. Prentice Hall PTR, 2002

[82] Herbert S. Wilf. *generatingfunctionology 3rd edition*. 2010

[83] Jack E. Graver. *Counting on Frameworks (mathematics to Aid the Design of Rigid*

Structures). 1993.

[84] Michael Soltys. *An introduction to the analysis of algorithms*. Published by World Sci-

entific 3rd edition 328 pages, 2018

- [85] Briland Hitaj, Paolo Gasti, Giuseppe Ateniese, Fernando Perez-Cruz *PassGAN: A Deep Learning Approach for Password Guessing* <https://arxiv.org/pdf/1709.00440.pdf>, 2019
- [86] Merkle R.C. *One Way Hash Functions and DES. In: Brassard G. (eds) Advances in Cryptology — CRYPTO' 89 Proceedings. CRYPTO 1989.* Lecture Notes in Computer Science, vol 435. Springer, New York, NY. https://doi.org/10.1007/0-387-34805-0_40, 1990