National College of Ireland

National College of Ireland

Project Submission Sheet

Student Name:	Matthew Browne		
Student ID:	x21174415@student.ncirl.ie		
Programme:	MSc/PGD in Cybersecurity	**Year:**	1
Module:	AI/ML in Cybersecurity (H9AIMLC)		
Lecturer:	Jaswinder Singh MSc/PGD		
Submission Due Date:	14th December 2025		
Project Title:	"How Can Machine Learning and Artificial Intelligence Be Used to Identify Malicious URLs, And How Does This Solve the Issue of a Safer Online Experience".		
Word Count:	7310		

I hereby certify that the information contained in this (my submission) is information pertaining to the research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the library. Using other authors' written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	Matthew Browne
Date:	12th December 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on the computer. Please do not bind projects or place in covers unless specifically requested.

4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI Acknowledgement Supplement

[AI/ML in Cybersecurity (H9AIMLC)

Your Name/Student Number	**Course**	**Date**
Name: Matthew Browne Student Number: x21174415	MSc/PGD in Cybersecurity	12/12/2025

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment
This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	**Brief Description**	**Link to tool**
Not Applicable		

Description of AI Usage
This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

Not Applicable

Evidence of AI Usage
This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

Not Applicable

Continuous Assessment Name	Part 2 Project,

	"How Can Machine Learning and Artificial Intelligence Be Used to Identify Malicious URLs, And How Does This Solve the Issue of a Safer Online Experience."
Student Name	Matthew Browne
Student ID	x21174415
Student Email	x21174415@student.ncirl.ie

"How Can Machine Learning and Artificial Intelligence Be Used to Identify Malicious URLs, And How Does This Solve the Issue of a Safer Online Experience".

Matthew Browne
MSc/PGD in Cybersecurity
E-mail *x21174415@student.ncirl.ie*

Abstract (100-150 words)

For my project looked at how I could use Artificial Intelligence (AI) and Machine Learning to help with and identifying URLs which could be perceived as normal but are actually malicious this helps individuals to stay secure, safe and protected online , some of the datasets I'll be using will be "Kaggle" [1] , "Phish Tank" [2] and the "UCI Machine Learning Repository" [3] These will all help me research and understand the practicality of using these datasets. My paper will investigate and research three different models "Random Forest" [4], "Convolutional Neural Network" [5] (CNN) for alongside (XGBoost) also known as Gradient Boosting [6].

For each of these, I'll be looking at how they can be used to screen and identify malicious links. Alongside this I'll be looking at metrics such as accuracy, precision recall and Auc. My result showed that "deep learning and ensemble methods performed much better than traditional techniques in recognizing harmful URLs." Through this I was able to show we both AI and ML can be used for an advantage in ensuring online safety which could in theory improve or even filter results , so on a wider scale if it was implemented at "ISP" level [6] we could ultimately protect millions of users from everyday harms such as phishing, fraud, identity theft and more.

I. Introduction

Phishing URL's such as ones we stumble across from sites and services we didn't sign up for , Malicious URL's such as ones we get from fake delivery companies and fraudulent links such as ones we get from real banks but with fake addresses have all become some of the most popular fundamental starting points for attackers to infiltrate our lives , everyday people from all walks of life and organization departments receive emails from unknown sources , messages on teams from internal and external vendors and often get tagged and sent requests through social media platforms like meta , WhatsApp and LinkedIn. Often unknown to the individuals and organizations they click on links they see in both there emails and social media posts these links are normally designed and purpose built to steal information like passwords, credit card information, or maybe even install harmful pieces of "spyware" [8] meaning that this information can be sent back unsuspectedly to an unknown entity with the intent of harming the individual.

With the nature of the threat landscape and the changes being faced minute by minute the traditional methods and tools used to prevent these kind of attacks are no longer sufficient , where previously we had "white lists" [*9*] and "black lists" [*10*] for blocking domains , websites and categorizing sites with "filtering" [*11*] these no longer provide for a sufficient protection mechanism , these are now considered slow and mundane with the rise of AI and ML attackers are becoming increasingly sophisticated with their methods of attack and we can no longer keep up using traditional older legacy techniques. Therefor to use them we would fall short of identifying malicious URLs with agility and speed, often resulting in lost time, damages and fraud to individuals and businesses.

With AI and ML, we can now look to incorporate them into our routine or preventative methods just like the "CIA Triad" [*12*] taught us which was to identify the "integrity" of a system, Ai and Ml can help us with trying to solve some of our more practical issues such as these which we have with URLs. By leveraging AI we can try to use it to recognize patterns it might see in different URLs , domains , and websites with the use of ML this can complement what AI can do by analyzing the URL patterns and behaviors essentially telling individuals and organizations if they pose a threat by hovering over or clicking on them before we actually do the action. With AI and ML together they can champion and recognize URLs which a known to be malicious in large datasets or repositories and make sense of there validity.

II. Motivation

Malicious URLS for some have been a common denominator with cyber-attacks they often serve as an entry point for bad actors' weather that's phishing, fraud, identity and credential theft and so on. Although there have been multiple stay safe online campaign runs warning individuals and businesses of the dangers, typical defense tactics are no longer enough to protect people from these attacks. The traditional methods of using blacklists, whitelisting domains, and filtering through dns servers none of these are methods are adequate to protect against the new types of malwares, spyware and ransomware that exist these days. As hackers are adopting new attack techniques and exploiting cve's this is achieved by hackers and nation state actors mimicking legitimate websites like your Meta, LinkedIn, Banking 365 and more by exploiting users with subtle changes to web address , bank websites look and feel , there able to trick users into entering their information they then through the use of AI and ML attack at speed to traverse across systems leaving little digital footprint. This shows that those older techniques which were once standard now have to be updated highlighting the need to change the way in which look at existing defenses which are now considered slow in response to threat actors, more reactive rather than proactive , speed and pace and which the detections are happening are often slower due to dated protocols and procedures this highlights the cyber risk involved in not knowing our threat landscape and perimeters, as a result of this more individuals and businesses are faced with deep consequences such as financial fraud, integrity of data , and availability of systems.

With the changes in how Artificial Intelligence and Machine Learnings are being implemented and developed they present both positives and negatives in our area of URL analysis where previously static methods had been employed with ML, we can now use this to learn from changes in the URL's being scanned. With ML we can leverage its ability to be able to learn from different identifiers and indicators in URLs making it easier to predict newer patterns by learning from the old ones. If this was to be included and leveraged as part of a Siem or Soc tooling we would be able to analyse our systems in real time, providing automatic classifications to the URL's and it would bring a value add as it would take away the mundane tasks and reduce the overall workload on analysts and investigators as the AI and ML would be working at scale to defend against attackers as part of security products and services. My motivation for

investigating and researching into this stems from my curiosity around the what if Ai and Ml could be used at scale to defend everyone against these URLs? Could it be built into the world wide web? Can we leverage it in security products like av products such as defender for endpoint, edr like defender for server, spam filters and more like spam titan. It also stems from the fact that systems are ageing, the practice of everything has changed and we as a society are going away from traditional on-premises to cloud based solutions like a Azure , Aws ,Gcp when I looked at using datasets like "Kaggle" , "Phish Tank" and "Uci Ml Repo" these allowed me to understand how we can train our Ai to understand false positives and false negatives this will help the Ai to be able to distinguish between what's right and what's wrong. testing out multiple model types on the URLs give us an understanding as to what models work better than others and which ones are more reliable, the models being "Random Forest", "XGBoost" and "CNN", bringing together our question "can AI And ML be used to identify malicious URLs" and is it effective enough. If it's the case it would highlight the need for strong legislation around having it implemented at the source such as ISP for safety reasons therefore mitigating the risk at ISP level rather than relying heavily on user interaction , this would also show how Ai and Ml again can add value to real world cases and prevent phishing attacks enhancing both individuals and business technology resiliency.

III. Literature Review

Due to the way in which hackers are getting and becoming more sophisticated and the attack methods their using phishing URL's have become more and more sophisticated this is why there has been a vast amount of interest across internet forums and IEE conferences around investigating the potential use of Ai and ML for detecting the URLs which are true and negative with a goal to identify these with speed and accuracy but also reliably e.g. [13] H. Le, Q paper. Earlier we spoke about how the more traditional methods like blacklisting domains and using indicators in signatures used to be sufficient this is no longer the case as they would heavily rely on previous patterns rather than analysing any of the unique characteristics in the URL themselves. With the paper I reviewed it honed in on how machine learning and deep learning could help and add value as it improves the accuracy in detecting good and bad URLs and the speed at which it does it , it does this by educating itself to understand the URLs like looking at its text it prides itself on being able to pick out features in the URL like the special characters , specific digits or the length of the URL , essentially doing it all without any human interactions.

Our eyes scan text on a page quickly this is known as "CNNs" the Ai and Ml use this same technique to be able to learn and look for patterns, this is of significant importance in terms of a URL like www.banking365.com an attacker may change the URL to something like www.banking365S3cure.com the technique looks for changes in the URL and differences this could be a letter or a number ether way they should look similar and because of this "URL Net" is able to produce better results in comparison with older counterparts preventative methods like scanners. , this tells me that the URL being analysed by the neural network spots some of the more tougher characteristic changes in comparison to what a human or scanner may see in this case it was easy but for others it's so subtle that it may not be seen. When looking at the study by (Saxena) this study looked at the elements of the URL itself and the accompanying features things like the text link name itself , server the URL was hosted on and the types of content on the webpage all of these elements were looked at to analyse and see where the bottlenecks were , this resulted in learning that if you just analysed the URL link , text name and the characters within the link there was no issues , but as soon you began to look into the information that accompanied the URL the task slowed down the whole process URL analyses becomes fast and is more efficient whereas analysing the content beyond the URL and its text resulted in loss of availability, this told me that if we were to use Ml we would need the model being used to be accurate , very fast without needing additional context or information from other web sources as after this is becomes unreliable.

It's understood from as least two of the papers I reviewed and probably more out there that a single model for Ml wouldn't be acceptable and it would be way under powered for the tasks and under pressure to deliver. Putting together models like "Random Forest and Gradient Boosting" together would bear more tangible results as a duo rather than each one separately, later I learned that this was known as an "ensemble" method what this does is it combines multiple models together which makes it much harder for a threat actor to be able to fool the detection mechanism , some of the research papers out there highlighted how attackers now try to fool the models into thinking the URL's are legit by replacing similar characters with characters that look very similar e.g. bankofireland.com bankofir3land.com or even g00gle instead of google most newer models can now detect these changes therefore warning the user. The literature also brought up some points which need to be clarified with the use of Ai in URL analysis, while it's good that models are able to identify when a URL is good or bad it's not ok for it to just identify them as one or the other , It should be providing reasoning around why a URL is identified as dangerous or malicious kind of like a prompt with a reason just like you would get if you had an issue with an application on your computer. With the investment into Soc and Siem tooling like Sentinel and IBM Qradar and other tools like lime these all-help analysts and hunters to understand why a threat had been identified and what characteristic or element is identified as being the indicator which flagged the block. This told me that by investigating and looking into how Modern Ai and Ml, models and techniques such as ensemble, deep learning and others I was able to grasp that these are much better than the traditional techniques and services being used to combat these attacks , pointing to a new realisation that traditional techniques will soon be on their way out replaced by a more automated and autonomous defence mitigation techniques going forward from vendors like Microsoft , IBM , Splunk and so on.

IV. Choice of Methods

The key question for my research was "How can Machine Learning and Artificial Intelligence be used to identify malicious URLs and to what extent can these techniques improve online safety in comparison with traditional methods" To outline and address the question I was asking I brought in three different Ml models which were

"Random Forest" , "Convolutional Neural Networks" and "XGBoost" all three models gave food for thought and insight into how different approaches could be used for the same outcome the assessment and testing of these approaches and models allowed me to understand more around the approaches some proving to have good performance others proving to be a bit redundant. The combination represents some traditional methods such as Ml, deep learning and ensemble. Each giving me a benchmark and a comparison where I could compare and contrast the "interpretability, accuracy and adaptability" of each of the approaches against one another. All of these being key points which were brought up across the different research papers.

(A) Random Forest

For random forest the scenario that best fit this was that if you had a team of purple red and blue team members looking at a URL and giving a verdict of whether it presented a risk to the organization or individual this is what random forest is. On person may look at the length of the URL, another will check if it has two many numbers or characters attached and then another will look for any weird symbols together each team member will provide their verdict it will there be a yes or a no. once the votes have been casted random forest will then aggregate them and return this vote which had the most answers , its heavily reliant on one answer over the other based on majority vote. I liked random forest because it was simple to under the decision process, it worked easily with URLs, it also gave a baseline to start from when I began to look at more advanced models. This allowed me to understand the models process in terms of a team vote and then how it picked out its definitive answer.

(B) Convolutional Neural Network

CNN works differently than random forest did in this instance you can describe it as being a single human who focuses on specific text features with their eyes rather than quickly scanning the text it looks at each character or number in a URL previously I called this out in my proposal where I said if we looked at something like bankofireland.com vs bank0fir3land.com both URLs are not the same the second one has an "0 and a 3" in place of the "o and the e", where's humans will spot this in some cases they won't always spot this that's where CNN takes a front seat it looks for the unsuspecting changes in a URL or name while humans can do this , sometimes it's not as fast or accurate in comparison to the way CNN does it. Although CNN does have a few standout qualities and abilities without human intervention, can learn and self-teach by itself, it's different to other models where others might miss characters and not see the patterns CNN does see them making it more reliable than other models out there. CNN has an advantage it's able to pinpoint and detect complex manipulations just like the ones that were mentioned in "URL Net Paper". This makes CNN a valuable ensemble unlike random forest and makes it more reliable in comparison with CNN's abilities to be able to notice the fragments of changes it makes it a priority model to use in phishing identification attacks.

(C) XGBoost

XGBoost is unique in terms of its capabilities when we looked at random forest being a group effort for evaluation, CNN being a Ml focused decision-making process, XGBoost brings into the mix a different lens its like random forest and a bit of CNN but with a twist if were looking at a url and we have the red , blue and purple team members if each one votes on the url , one says No the other says yes and the last one says yes , if the result is the answer is yes and the first person got it wrong it's the duty of the second person to go back and fix the first persons answer , the third person then just learns from both the first and the second person with all the learnings from each individual the whole premise is that the system for choosing the correct answer learns as a lessons learned exercise with that new learning it brings that into the next decision or question. From the investigation I was able to see that it works well when the ratio bad to good URLs is higher, it's also able to reason better than random forest and it learns from its mistakes, what makes it a good choice is that it's fast and suited to Realtime detection.

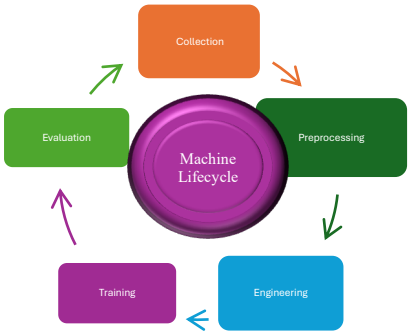
(D) Why All Models

Model Name	Reason	Model Type
Random Forest	Valuable when used for basic checks	Traditional
CNN	Valuable when used to spot differences	Deep Learning
XGBoost	Valuable when you need an all-round solution (very fast to check, self-checks itself, can handle large data sets)	Machine Learning

The key pitfalls from the literature reading were that a single model wouldn't be of any benefit when accessing links , this is majority because it would like the time , attention to detail and the ability to be able to access links properly while also learning from them , if you used a model like CNN or XGBoost it provides better results due to their infused ensemble configurations , random forest looks like a model which should only be used for checks which are foundational rather than anything that's specialty or expert level. It also highlighted that in the future models like random forests may become redundant due to the fact that it only performs basic functionality and doesn't actually learn from itself, this is a big no for future requirements meaning it wouldn't be a contender in a idr (intrusion detection system) or ips (intrusion prevention system).

V. Methodology

For my methodology I went through a set of phases to begin and carry out the work required the process was based of the more in depth "Machine Learning Lifecycle" [14] but just streamlined for a quicker process overview.

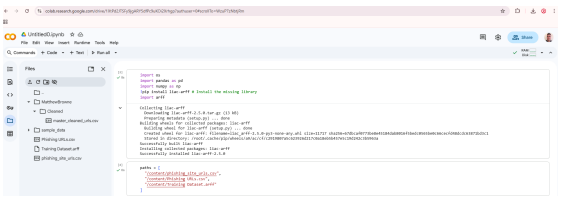


Key Commands Used

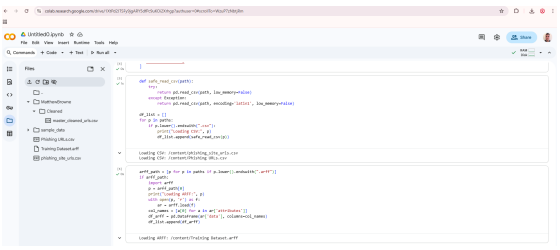
Command Name	Command String	Command Reference
Read	(pd.read_csv()).	Panda Read
Drop	df.drop_duplicates()	Panda Drop

1. The Collection Phase

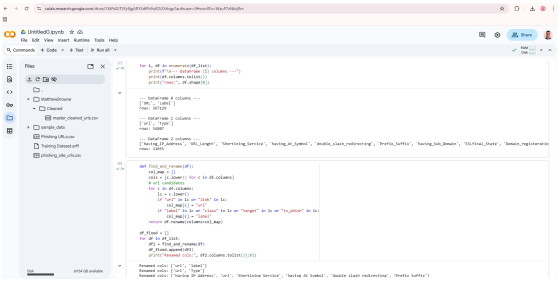
My steps involved downloading three different data sets from publicly available sites such as “Kaggle” “Phish Tank” and “UCI Repository” I saved each of these into a folder called MatthewBrowne/datasets



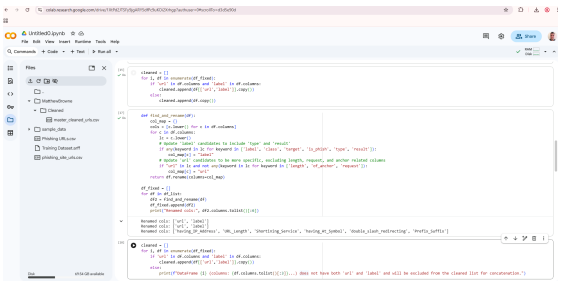
I then loaded each data set into “Pandas” after which I merged each data set into a larger data frame after this, I proceeded to remove any duplicates leaving me with a singular master data sheet.



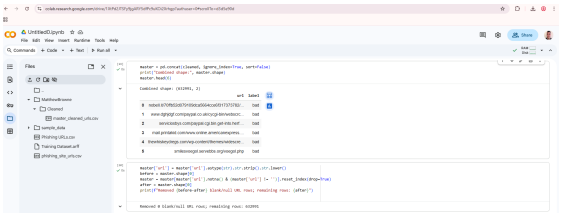
I then loaded the locations and files



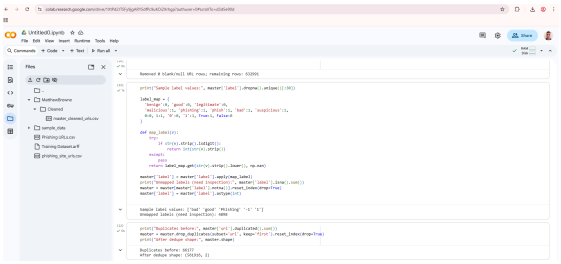
I then loaded csvs and ensured there was no coding issues



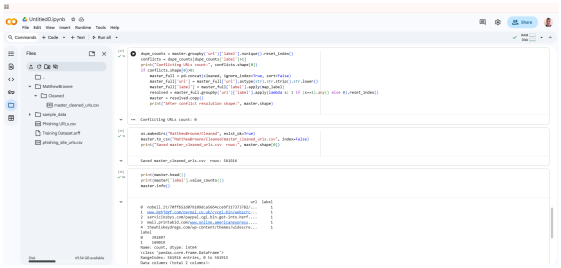
With the Arff file I had to get this converted into a pandas data frame to be able to use it.



I differentiated and identified which columns contained the URLs and which ones contained labels



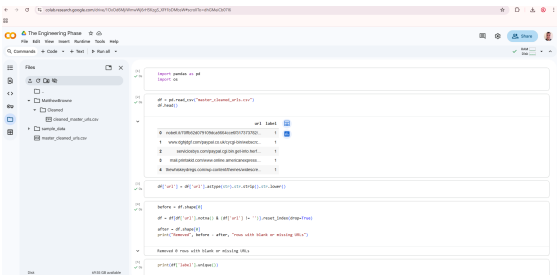
After that I renamed the columns to some common standards such as URL and label, ensuring labels had a numeric value I also checked to ensure that there were no duplicate URLs.



I marked Urls as suspicious if the sources pointed to them being suspicious via indicators.

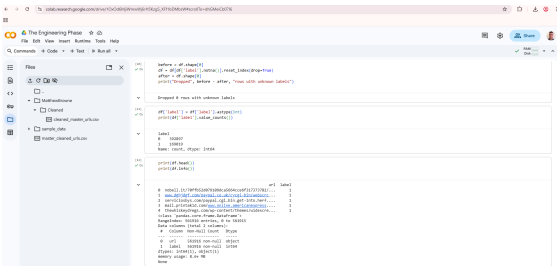
2. The Pre-processing Phase

I loaded each data set into “Pandas” after which I merged each data set into a larger data frame after this I proceeded to remove any duplicates leaving me with a singular master data sheet.



I opened pandas so that I could clean it , then I ensured that all my urls were lowercase alongside checking if I had rows with urls missing making sure that there were no blank entries here , I also checked how the labels were showing in the file mapping back the relevant label 0 shows as benign and 1 showing as malicious.

From here I created the different mapping types again 0 being legitimate and 1 being bad I did this so that the machine learning could understand the numerical formats I then checked and confirmed that all texts were showing as 0 and 1 ensuring any labels which were different were mapped back correctly.

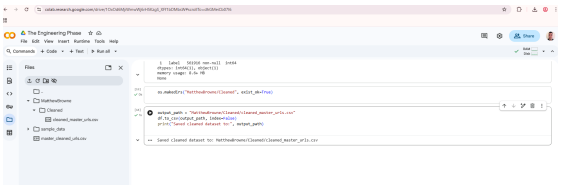


I did a final check, exported the cleaned dataset for my training phase and then saved it into my folder.

3. The Engineering Phase

I began converting all URLs to lowercase characters, ensured there were no blank entries or fields, then I renamed the labels, so they matched with 0 being benign and 1 being malicious afterwords I stored the clean data sets in MatthewBrowne/Cleaned

I removed rows which had no labels as they weren't going to help in any way remembering also to ensure that my labels were integers such as 0 and 1 effectively keeping them clean



4. The Training Phase

My Steps for split the data was divided into

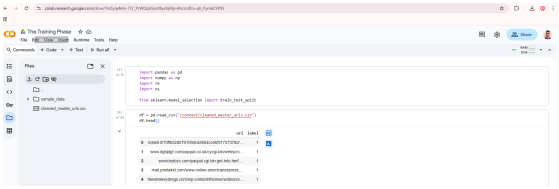
- 70% training,
- 15% for validations
- 15% Testing,

_____
100% total allocated

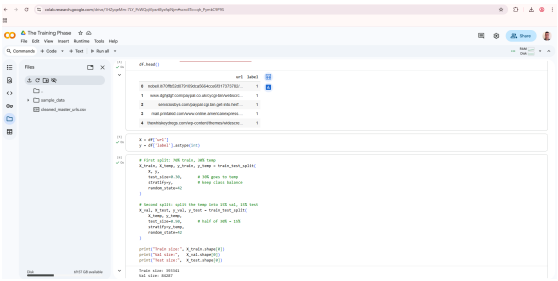
Next, I went onto training each of the models because each model is different, I was required to train them, the research papers also indicated to me that there isn't a simple perfect model which works so by comparing I was able to see this for myself this helped with addressing my research question. So, for each model they were trained separately I did this because I wanted to ensure that each model got the right inputs to give me out the correct results this is because each model understands URLs differently. I also wanted to see if I could put in different detections for the models and then compare the results for each one. Lastly, I want to answer my research question based on the results essentially pointing me to which Model was best and can it actually help keep individuals and organisations safe online.

For each Model I created what's called "different features" for both Random Forest and XGBoost because their traditional machine learning models they needed numbers as they are unable to read text like humans to be able to read these properly, I converted them into numerical values. This would then have helped the model to be able to digest and understand things like patterns, attacker tricks and the risk levels allowing the models to understand the information properly.

While CNN's on the other hand and are different , they can unlike Random Forest and XGBoost read the characters in the raw text files , for CNN'S you need to have a fixed length a number value and a format which is structured to be able to do this you need to put each URL into characters put these characters into number values , prep them like you would as read in the "URL Net" paper doing all of this allows you to apply pattern recognition but at the deep learning level. By doing and testing all three models I'm able to formulate a proper response to my research question backed by data analytics and evidence.

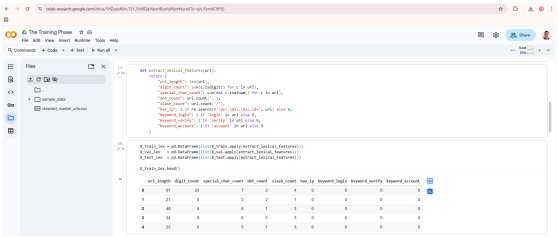


Next, I needed to load my previous clean dataset

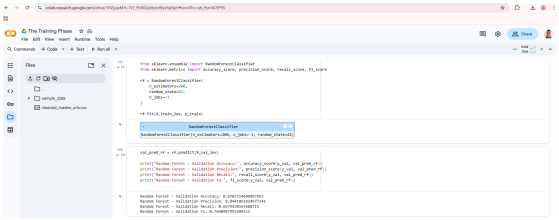


Splitting each of the sets into

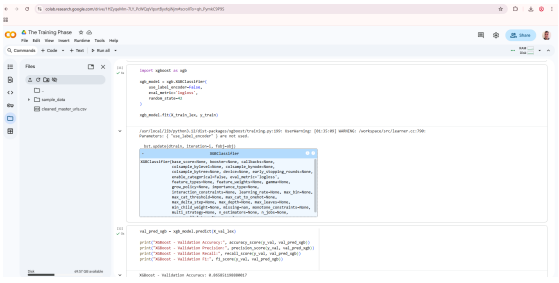
1. 70% Training
2. 30%, with 15% Validation + 15% Testing



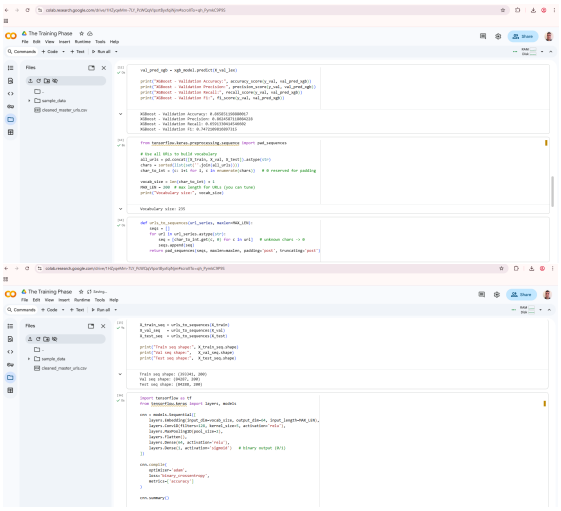
Moving on to creating different features and functions for both Random Forest and XGBoost, as these models cannot read text files there urls need to be converted into numbers / converting urls to padded sequences



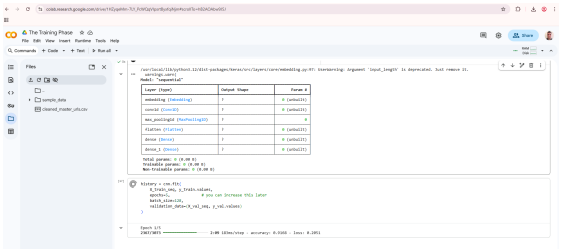
After that I moved onto training the validation and test urls, I did this because I just converted the urls into numbers that describe different characteristics such as length , digit quantity , special characters and more , what this does is it helps XGBoost to see the different types of patterns attackers use this is then what makes it perfect to feed into CNN



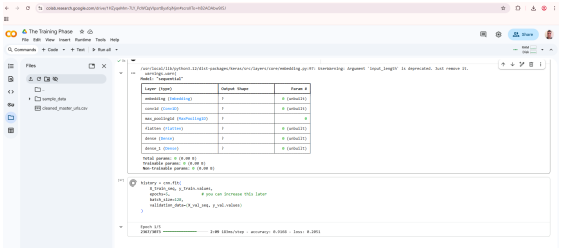
I moved onto training the random forest baseline

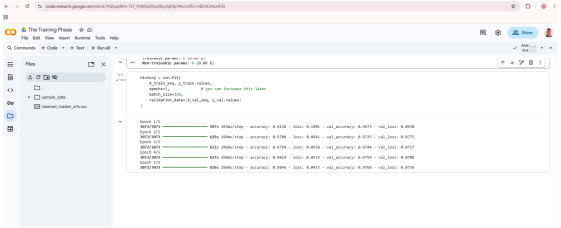


Quickly performing a sanity check, this allowed me to perform a comparison based on the baseline which I had earlier.

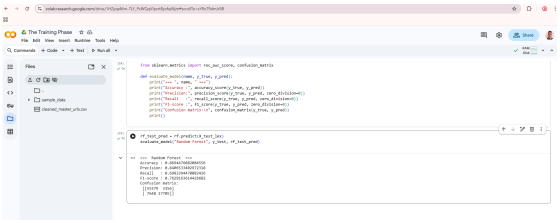


As XGBoost is a more advanced version of tree-based learning, it adapted and learned from its mistakes from this I built and trained the CNN

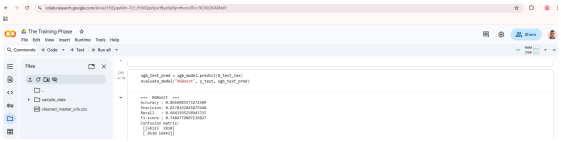




This process was slower than I expected here I began to build out and train the CNN , we know from the “URLNet” research paper that this model learns from the characteristics.

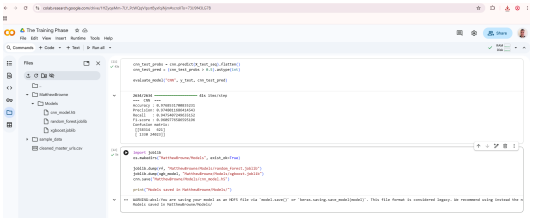


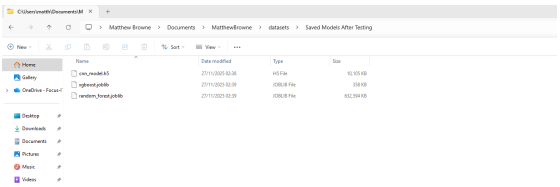
Once the process was completed, I moved onto testing the performance first up was the Random Forest test



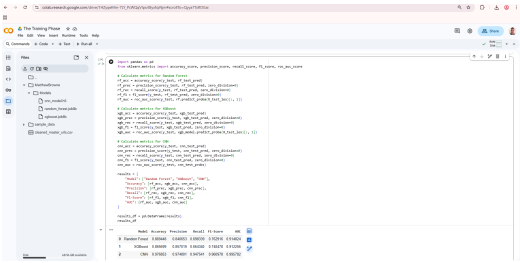
After Random Forest testing concluded I tested XGBoost

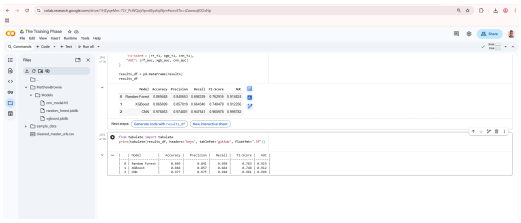
I did it this way because I wanted to test each one fairly and it ties back to the question Does AI/ML actually help detect malicious urls and is there a difference with CNN





Once my testing was completed, I saved copies of my models





After this I exported the results in a comparison table

The Applications I learned about as part of this

App Name	Purpose	App Link
Jupyter	cooding	Jupyter
Python	Pre-Req	Python
Pandas	Loading/cleaning data	Pandas
NumPy	Pre-Req	Numpy
Scikit-Learn	Train/Metrics	Scikit-Learn
XGBoost	Pre-Req	XGBoost
TensorFlow	For CNN Pre-Req	TensorFlow
Matplotlib	For Graphs	Matplotlib

The Datasets I used to perform the testing

Dataset Name	Dataset Site
Kaggle	Phishing Site URLS
Mendeley	Mendeley Urls
UCI ML Repository	UCI Phishing Site URLS

Videos I Used to Understand the more about what was being asked

"Create, transform and select features." [25]	https://youtu.be/gf2bfjn2s4o
"Feature engineering for text data." [26]	https://youtu.be/EvfnDphF-9g
"Why feature engineering matters" [27]	https://youtu.be/j-kkH3q0hLE
"Character-level URL/text modelling" [28]	https://youtu.be/m75dVDddymU
"Practical implementation of character-based CNN." [29]	https://youtu.be/CNY8VjJt-iQ
"CNNs for text data." [30]	https://youtu.be/8YsZXTpFRO0
"Traditional ensemble methods" [31]	https://youtu.be/wO7YBNPCu58

(A) Random Forest

I wanted to start with something simple; I needed a baseline to begin the process it was going to show me how well it worked and allowed me to compare the base model with other more advanced models like XGBoost and CNN.

The Result

```
[19]
✓ 5s
rf_test_pred = rf.predict(X_test_lex)
evaluate_model("Random Forest", y_test, rf_test_pred)

===  Random Forest  ===
Accuracy : 0.8694476082004556
Precision: 0.8406533402972318
Recall   : 0.6983394470082436
F1-score : 0.7629163614426682
Confusion matrix:
 [[55579  3356]
 [ 7648 17705]]
```

(B) XGBoost

Now that I had got comfortable with simpler baseline model, I wanted to see how a more advanced model would handle wrong inputs, also wanted to see did it perform better then random forest, and I also wanted to see the speed at which it did it.

The Result

```
[20]
✓ 0s
xgb_test_pred = xgb_model.predict(X_test_lex)
evaluate_model("XGBoost", y_test, xgb_test_pred)

===  XGBoost  ===
Accuracy : 0.8656985573272589
Precision: 0.8570192845875948
Recall   : 0.6643395258943715
F1-score : 0.7484779807136827
Confusion matrix:
 [[56125  2810]
 [ 8510 16843]]
```

(C) CNN

Now that I had got an understanding of how both the baseline "random forest" and intermediate "XGBoost" performed CNN was the opportunity to see how well it did perform with more advanced detections and more trickier URLs, proving its abilities with this essentially told me whether or not "AI and Ml Could detect Malicious URL's"

The Result

```
[21]
✓ 43s
cnn_test_probs = cnn.predict(X_test_seq).flatten()
cnn_test_pred = (cnn_test_probs > 0.5).astype(int)

evaluate_model("CNN", y_test, cnn_test_pred)

2634/2634 ━━━━━━━━━━━━━━━━━━━━ 41s 15ms/step
===  CNN  ===
Accuracy : 0.9768531700835231
Precision: 0.9748011686414543
Recall   : 0.9475407249635152
F1-score : 0.9609776586595196
Confusion matrix:
 [[58314   621]
 [ 1330 24023]]
```

Comparison Of Results

Model Type	Model Accuracy	Model Precision	Model Recall	Model Score	Auc
Random Forest	0.869	0.841	0.698	0.763	0.915
XGB	0.866	0.857	0.664	0.748	0.912
CNN	0.977	0.975	0.948	0.961	0.996

Some of the key questions after going through the process and testing was to understand how the Ai/Ml models compared linking this back with my research question to give a definitive Answer.

1. Which Ai/Ml model performed adequately?

   Based on the results it could be said random forest returned an adequate result although not by much and not high performing.

2. Which Ai/Ml model showcased the most promising result?

   The CNN model showed me the best result by a big margin outperforming both other models.

3. Did any Ai/Ml model make too many mistakes?

   Both Random Forest and XGBoost made considerable errors and couldn't be considered as reliable.

4. Which Ai/Ml model provided the best Defence against attackers?

   From the results CNN did the best here as I could see its recall was 94.8% meaning it missed fewer then 6% of results and its Auc showed a near perfect score meaning it could confidently distinguish between good and bad urls making it a primary use case for filtering.

5. Is deep learning better then machine learning?

   Yes, as it outperformed across the board meaning that it was the best at identifying patterns in in comparison to both Random Forest and XGBoost.

6. Was there any improvement against detections by using XGBoost?

   No Notable improvements outside of the precision capabilities and that was just minor.

7. Is the baseline for Random Forest sufficient?

   Absolutely not no it couldn't be relied upon as a secure baseline as it misses to many urls and would be a security risk in an organisation.

8. Which Ai/Ml model provided the best protection for a real-world scenario?

   CNN Provided the best detections and preventions for use in a real-world scenario, this tells me that CNN would be the best suited to being implemented by an ISP & Soc Provider.

VI. Evaluations

When looking at the three models and cross comparing them with Random Forest , it performed as expected , it very much stood as a baseline but not the best baselines XGBoost made allot of errors as it didn't sufficiently identify a significant quantity of the URLs , also in comparison with Random Forest XGBoost didn't outperform , which tells us both wouldn't be good in a real world scenario. When I looked at CNN, I was able to see it beat both Random Forest and XGBoost . This tells me

that there is a need for deep learning in filtering systems, it's essentially superior to the other models if this was deployed in a real-world scenario for a ISP it would catch a lot of false URLs.

Conclusion

The purpose of my paper was to understand and evaluate whether Artificial Intelligence (AI) and Machine Learning (ML) or even Deep Learning could effectively identify and detect and respond to good and bad URLs this was determined by identifying which Models could provide the most benefit in terms of protection for organisations and individuals. As part of the study, I incorporated Random Forest, XGBoost and CNN each providing different functionalities, Random Forest was used as the standard baseline, XGBoost was used as the "Ensemble" Convolutional Neural Network was used as the deep learning, how I assessed each of these were through stress points like accuracy , precision , recall F1-Score and AUC each of these models went through these test with the datasets all models were exposed to unseen URLs.

The results at the end were surprising in terms of the baseline and intermediate level models but the deep learning model came out significantly higher, this showed that there were tangible differences between Traditional ML such as Random Forest Vs CNN which is deep learning. Random Forest turned out a good score for a traditional model giving me a 86% accuracy and a good precision score this tells me that it worked as a baseline in terms of identifying the bad urls but at the same time its recall score of 0.698 tells me that it wouldn't be reliable in a real world scenario , this is because it would be two prone to missed threats which could have adverse effects on an organization. XGBoost the second model offered me some improvement over Random Forest this was seen primarily in its precision score which was 0. 857.Again, XGBoost had had an even worse recall value making it an unreliable solution also in a real-world scenario as it just missed worse urls then the baseline did. With the Dataset presented XGBoost wasn't able to articulate some of the URL's correctly this tells me that it didn't handle some of the characters or tricks which were in the URL's this essentially meant that it didn't provide any benefit as it still made too many missed detections and wouldn't be useable in a production environment.

When I looked at the third model CNN it should stronger results to both Random Forest and CNN, with CNN scoring high across the board it showcased its abilities to outperform the others and distinguish itself from the other model it was able to identify between good and bad urls with rapid speed, alongside this it was also able to give the best recall score this was important as the recall allows it to identify the bad URLs in their majority. This allowed me to validate the research from "URL Net" this brought me into the world of deep learning models and how there able to detect character level patterns that attackers put into the URL's , where Random Forest and XGBoost rely solely on numerical values , CNN learns from the strings this empowers it to be able to pick up on the little details which other models cannot detect.

After going through the process from a cyber point of view CNN must be the clear winner here, that high recall value distinguishes it as a leader telling me that it doesn't really misclassify threats this is important in a world where systems are providing false negatives and not identifying threats correctly, this is what makes CNN essentially a viable deployment option for ISP's and Soc's to integrate into their filtering programs. CNN caters to organizations and individuals who care about things like finance theft, credential theft or breaches; traditional ML options wouldn't be suited for high level protection the way CNN can do it. Due to the dependency on Deep learning like CNN traditional ML like Random Forest and XGBoost wouldn't survive in the cybersecurity realm as it just not a high-performance option, traditional MLs just don't have the power to be able to perform the way in which CNN's do.

The difference between the 3 options is night and day, while Random Forest does provide a baseline it still isn't able to perform at the levels CNN does, XGBoost is the same while it introduces some extra resiliency in comparison to Random Forest, its benefits are outweighed by its low score in other areas. CNN was able to identify all the different components and structures of the Urls whereas Random Forest and XGBoost couldn't do that this is a clearer indicator that deep learning architectures are more suits to classification with URLs , this is especially true where pattern matching is key.

As I revert to looking and thinking about the research and weather organisations and individuals use AI/ML to improve the detections the research and analysis of data sets tell me it's a strong yes but specifically for Deep Learning. The other two models just don't show as much of a promise as CNN does and don't have the capabilities to provide the best coverage where security gaps are concerned, CNN is practical and is suits for real world deployments, if this was implemented at ISP Levels this could and would reduce things like phishing attacks, identity theft and fraud by being able to identify the good and bad urls before someone clicks them this could provide for a practical solution to some of the world's biggest issues around fraud.

In short, the research tells me that traditional ML models could be used to serve as a baseline for more advance level deep learning models within the Url Detections where and if required, while all two models incorporate and learn from each other in particular Ensemble and Deep Learning the Random Forest model would only serve as a base model and nothing more. With that being said it should be acknowledged that Deep learning like CNNs provide the most effective way to detect malicious, bad and good URLs because of this CNN offers individuals and organizations the most effective real-world solution for identifying URLs this means that AI can essentially provide us with a meaningful enhancement for online safety if implemented correctly therefore answer my research question.

VIII References

[1] "PhishTank | Join the fight against phishing," www.phishtank.com. https://www.phishtank.com/ (accessed Nov. 1st, 2025).

[2] Kaggle, "Kaggle: Your home for data science," Kaggle.com, 2025. https://www.kaggle.com/ (accessed Nov. 1st, 2025).

[3] "UCI Machine Learning Repository," archive.ics.uci.edu. https://archive.ics.uci.edu/ (accessed Nov. 1st, 2025).

[4] IBM, "What is random forest?," Ibm.com, Oct. 20, 2021. https://www.ibm.com/think/topics/random-forest (accessed Nov. 2nd, 2025).

[5] IBM, "Convolutional Neural Networks," Ibm.com, Oct. 06, 2021. https://www.ibm.com/think/topics/convolutional-neural-networks (accessed Nov. 2nd, 2025).

[6] E. Kavlakoglu and E. Russi, "XGBoost," Ibm.com, May 09, 2024. https://www.ibm.com/think/topics/xgboost (accessed Nov. 2nd, 2025).

[7] Wikipedia Contributors, "Internet service provider," Wikipedia, Aug. 07, 2019. https://en.wikipedia.org/wiki/Internet_service_provider (accessed Nov. 2nd, 2025).

[8] Wikipedia Contributors, "Spyware," Wikipedia, Feb. 27, 2019. https://en.wikipedia.org/wiki/Spyware (accessed Nov. 2nd, 2025).

[9] "Whitelist," Wikipedia, Sep. 06, 2023. https://en.wikipedia.org/wiki/Whitelist (accessed Nov. 2nd, 2025).

[10] "Blacklist (computing)," Wikipedia, Feb. 11, 2021. https://en.wikipedia.org/wiki/Blacklist_(computing) (accessed Nov. 2nd, 2025).

[11] "Internet filter," Wikipedia, Oct. 29, 2021. https://en.wikipedia.org/wiki/Internet_filter (accessed Nov. 3rd, 2025).

[12] Fortinet, "What is the CIA Triad and Why is it important?," Fortinet, 2025. https://www.fortinet.com/resources/cyberglossary/cia-triad (accessed Nov. 3rd, 2025).

[12] "\r\n \n Integrity-Based Cyber Attacks Against AI Systems\n \r\n," Threatintelligence.com, 2022. https://www.threatintelligence.com/blog/integrity-based-attacks-ai (accessed Nov. 3rd, 2025).

[13] H. Le, Q. Pham, D. Sahoo, and S. Hoi, "URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection," 2018. Available: https://arxiv.org/pdf/1802.03162 (accessed Nov. 3rd, 2025).

[14]GeeksforGeeks, "Machine Learning Lifecycle," GeeksforGeeks, Mar. 27, 2024. https://www.geeksforgeeks.org/machine-learning/machine-learning-lifecycle/ (accessed Nov. 4th, 2025).

[15]R. Mistry, "Data Splitting (Train-Test-Validation) in Machine Learning," Medium, Feb. 14, 2025. https://medium.com/@rohanmistry231/data-splitting-train-test-validation-in-machine-learning-2d5d1927fa69 (accessed Nov. 4th, 2025).

[16] Wikipedia Contributors, "Stratified sampling," Wikipedia, Feb. 17, 2019. https://en.wikipedia.org/wiki/Stratified_sampling (accessed Nov. 4th, 2025).

[17] "Pandas Read CSV," www.w3schools.com. https://www.w3schools.com/python/pandas/pandas_csv.asp (accessed Nov. 4th, 2025).

[18] "Pandas DataFrame drop_duplicates() Method," www.w3schools.com. https://www.w3schools.com/python/pandas/ref_df_drop_duplicates.asp (accessed Nov. 5th, 2025).

[19] Pavel Surmenok, "Character-level Convolutional Networks for Text Classification," Medium, Jun. 13, 2016. https://medium.com/@surmenok/character-level-convolutional-networks-for-text-classification-d582c0c36ace (accessed Nov 5th, 2025).

20] A. Lev, "XGBoost versus Random Forest | Qwak's Blog," www.qwak.com, Dec. 19, 2022. https://www.qwak.com/post/xgboost-versus-random-forest (accessed Nov. 5th, 2025).

[21] A. Lev, "XGBoost versus Random Forest | Qwak's Blog," www.qwak.com, Dec. 19, 2022. https://www.qwak.com/post/xgboost-versus-random-forest (accessed Nov. 5th, 2025).

[22] "Phishing Site URLs," www.kaggle.com. https://www.kaggle.com/datasets/taruntiwarihp/phishing-site-urls (accessed Nov. 6th, 2025).

[23] "PhishTank | Join the fight against phishing," www.phishtank.com. https://www.phishtank.com/ (accessed Nov. 6th, 2025).

[24] "UCI Machine Learning Repository," archive.ics.uci.edu. https://archive.ics.uci.edu/dataset/327/phishing+websites (accessed Nov. 6th, 2025).

[25]Youtu.be, 2025. https://youtu.be/gf2bfjn2s4o (accessed Nov. 10th 2025).

[26]Youtu.be, 2025. https://youtu.be/j-kkH3q0hLE (accessed Nov. 10th 2025).

[27]Youtu.be, 2025. https://youtu.be/j-kkH3q0hLE (accessed Nov. 10th, 2025).

[28]Youtu.be, 2025. https://youtu.be/m75dVDddymU (accessed Nov. 11th, 2025).

[29]Youtu.be, 2025. https://youtu.be/CNY8VjJt-iQ (accessed Nov. 11th, 2025).

[30]Youtu.be, 2025. https://youtu.be/8YsZXTpFRO0 (accessed Nov. 11th, 2025).

[31]Youtu.be, 2025. https://youtu.be/wO7YBNPCu58 (accessed Nov. 12th, 2025).

[32] Wikipedia Contributors, "Feature model," Wikipedia, Aug. 05, 2025. Feature model - Wikipedia (accessed Nov. 12th, 2025).