Neural Network Analysis on the Phishing Dataset

Theodore Fitch
Department of Data Analytics, University of Maryland Global Campus

DATA 630: Machine Learning

Dr. Ami Gates

July 20th, Summer 2021

Introduction:

With software becoming more advanced every day, phishing attacks are becoming more sophisticated and numerous as well. As defined by the APWG (2021):

> "Phishing is a crime employing both *social engineering* and *technical subterfuge* to steal consumers' personal identity data and financial account credentials."

Social engineering is when criminals convince victims that they are from a legitimate source. A common tactic has been to impersonate a Microsoft employee attempting to prevent a virus from taking over your computer or needing to make "critical" updates to a laptop. An unsuspecting victim will then allow administrative rights to their computer to the hacker allowing the hacker to take what they want. Hackers will often impersonate common, well-trusted brands in order to put their victims at ease when they introduce themselves.

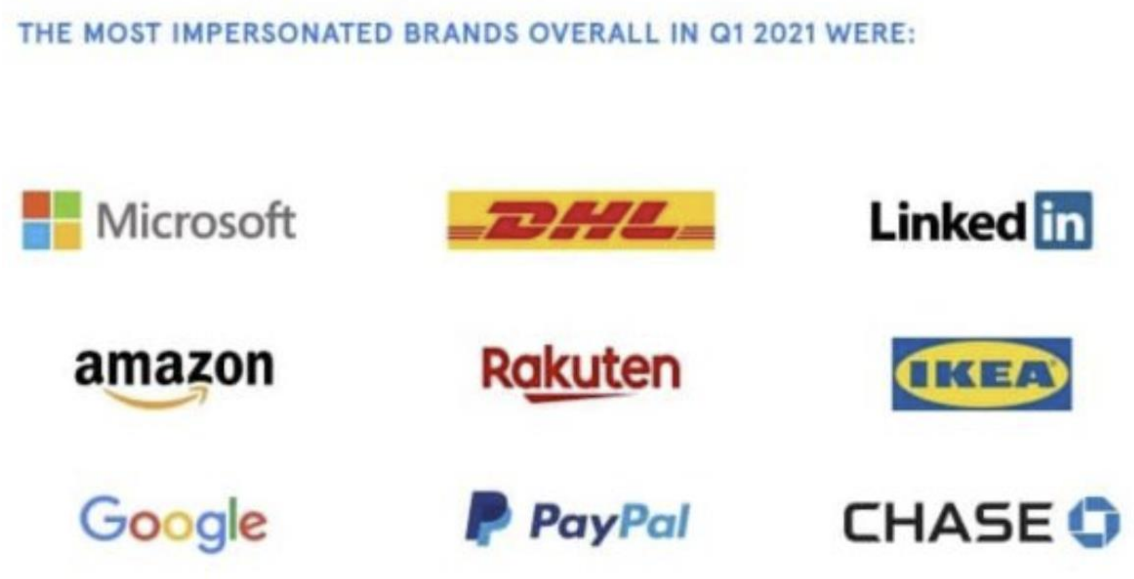


Figure 1. Attackers will often impersonate common, well-trusted brands in order to gain access, take data, or extort money (Tessian, 2021).

The “technical subterfuge” involves countless tricks for modern phishers. Links are truncated to hide malicious domain names or links are extended to absurd lengths to obscure the malicious domain within it. This way, a victim may only read the first part of a link and think that it is safe. Phishers may use a non-malicious host website to automatically redirect victims to a malicious website. They may also project an image of a legitimate website on top of a malicious website. There are nearly countless tricks a criminal has at their disposal to illegitimately garner information. This combined with social engineering is a potent combination. Users will interact with a person thinking they are from a valid source. Then, the hacker will use technical abilities to make the person think they are doing the right thing to get help when really, they are giving the hacker access to restricted information.

Why do phishers do it? The average cost of a data breach is $150 per record and $3.86 million overall for Personally Identifying Information (PII) which is the most common type of breach (IBM, 2021). At that rate, a data breach of only 1,000 records would yield damage control of $15,000. For companies that were working remote due to COVID-19, the average cost of a data breach increased from $3.86 million to $4 million (IBM, 2021). Not only do companies experience an upfront cost to data breaches, but they also tend to see a stock price drop of 5% (Tessian, 2021). The costs include: lost hours from employees; remediation; incident response; damaged reputation; lost intellectual property; direct monetary losses; compliance fines; lost revenue; and legal fees (remediation is the largest of these categories claiming most of the damage quantitatively)(Tessian, 2021). While many crimes focus on selling a product which can only be sold once, phishing is relatively inexpensive for the criminal. They can use the same programs over and over. Thus, phishing is a lucrative crime with little overhead.

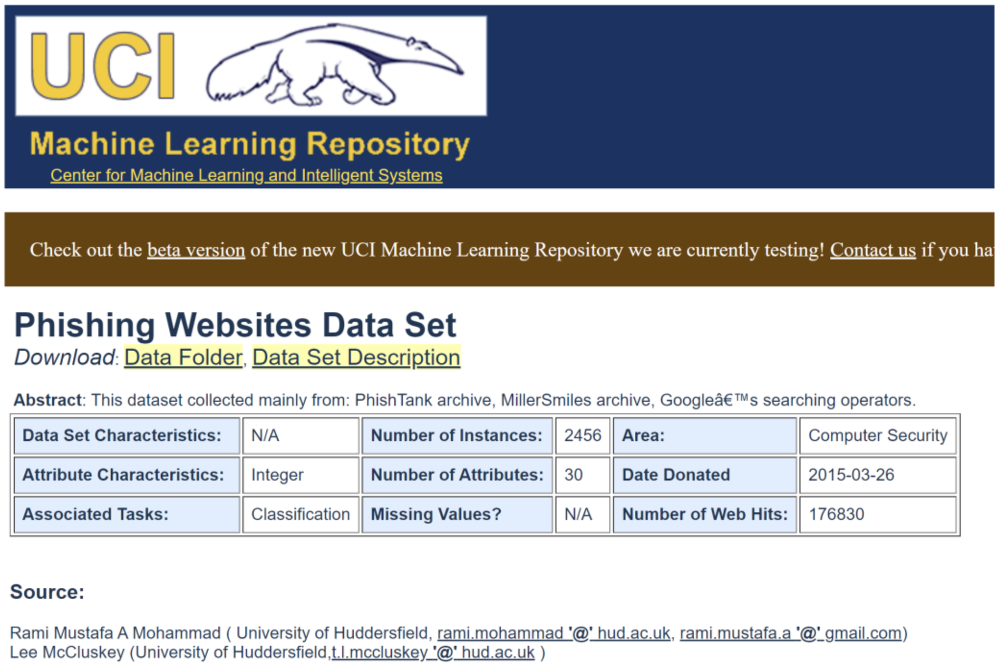
Phishing is ubiquitous now. Most people who have a work email account have likely received many phishing emails attempting to gain access or information. It is the most common type of cybercrime now with 11 times more reports in 2020 as 2016 (Tessian, 2021). Records were shattered last year where attacks had more than doubled in the 4th quarter of 2020 versus 4th quarter 2020 of reported global phishing attacks (APWG, 2021). While many crimes require criminals to interact with other people in person, phishers can take a "shotgun approach" sending many phishing attempts at once. Then, if an unsuspecting victim begins to take the bait, the phisher can invest more heavily.

There has been significant work in the field of Spam and phishing prediction. Many phones nowadays come preloaded with software to predict phishing texts, calls, emails, and other messages. Most email and other messaging apps also contain safeguards against phishing. The current mean susceptibility rate is approximately 21% (Sommestad & Karlzén, 2019). This can be greatly influenced negatively or positively. Susceptibility can be reduced via trainings, company awareness, and phishing detection software; but it can be increased based on the type of phishing attempt used, how complicated and well designed the attack is, and how personalized the attack is (Sommestad & Karlzén, 2019). If companies want to survive the onslaught of inevitable phishing attacks, they will need to implement software which can detect phishing.

Analysis and Model Demonstration:

Data Information:

This dataset was procured from the UCI website (UCI Machine Learning Lab, 2015). The website does not list whether it is generated artificially or generated from real-world data. It was in CSV format. It shall be referred to as "PH" henceforth for the sake of brevity.


UCI
Machine Learning Repository
Center for Machine Learning and Intelligent Systems

Check out the beta version of the new UCI Machine Learning Repository we are currently testing! Contact us if you ha

Phishing Websites Data Set

Download: Data Folder, Data Set Description

Abstract: This dataset collected mainly from: PhishTank archive, MillerSmiles archive, Googleâ€™s searching operators.

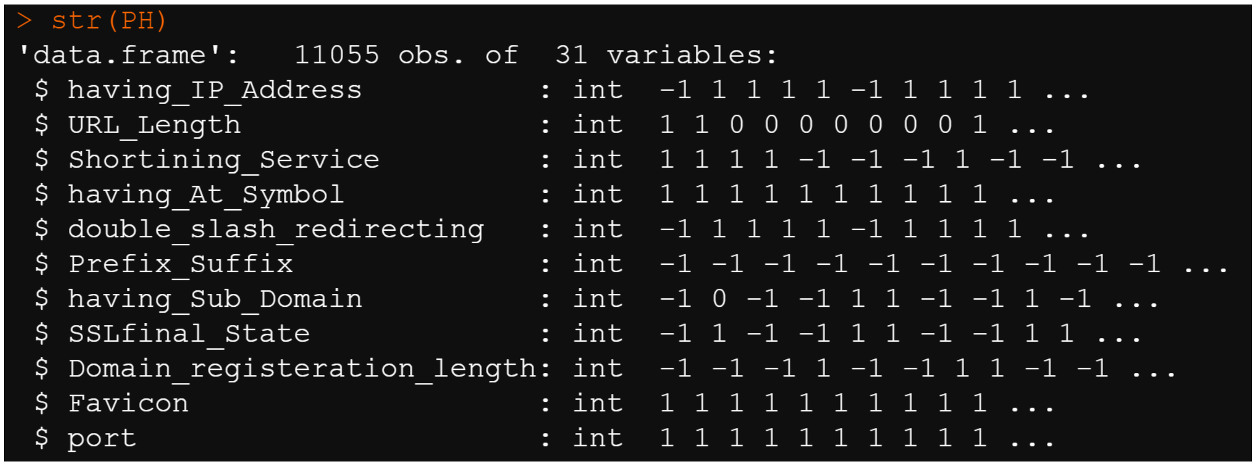
Data Set Characteristics:	N/A	**Number of Instances:**	2456	**Area:**	Computer Security
Attribute Characteristics:	Integer	**Number of Attributes:**	30	**Date Donated**	2015-03-26
Associated Tasks:	Classification	**Missing Values?**	N/A	**Number of Web Hits:**	176830

Source:

Rami Mustafa A Mohammad (University of Huddersfield, rami.mohammad '@' hud.ac.uk, rami.mustafa.a '@' gmail.com)
Lee McCluskey (University of Huddersfield,t.l.mccluskey '@' hud.ac.uk)

Figure 2. The UCI website shows the summary of the Phishing Websites Data Set.

Exploratory Data Analysis:



```
> str(PH)
'data.frame':	11055 obs. of  31 variables:
 $ having_IP_Address          : int  -1 1 1 1 1 -1 1 1 1 1 ...
 $ URL_Length                 : int  1 1 0 0 0 0 0 0 0 1 ...
 $ Shortining_Service         : int  1 1 1 1 -1 -1 -1 1 -1 -1 ...
 $ having_At_Symbol           : int  1 1 1 1 1 1 1 1 1 1 ...
 $ double_slash_redirecting   : int  -1 1 1 1 1 -1 1 1 1 1 ...
 $ Prefix_Suffix              : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
 $ having_Sub_Domain          : int  -1 0 -1 -1 1 1 -1 -1 1 -1 ...
 $ SSLfinal_State             : int  -1 1 -1 -1 1 1 -1 -1 1 1 ...
 $ Domain_registeration_length: int  -1 -1 -1 1 -1 -1 1 1 -1 -1 ...
 $ Favicon                    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ port                       : int  1 1 1 1 1 1 1 1 1 1 ...
```

Figure 3. Using the command Structure on PH showed there are 31 variables and 11,055 rows of data (only 11 variables are shown in this image).

PH contained 31 variables with 11,055 rows of data. The variable of interest, “Result”, represents whether a result is indeed Phishing or is not Phishing. All variables are factors with 2 to 3 levels with 1s indicating something is less likely to be Phishing, -1s indicating something is

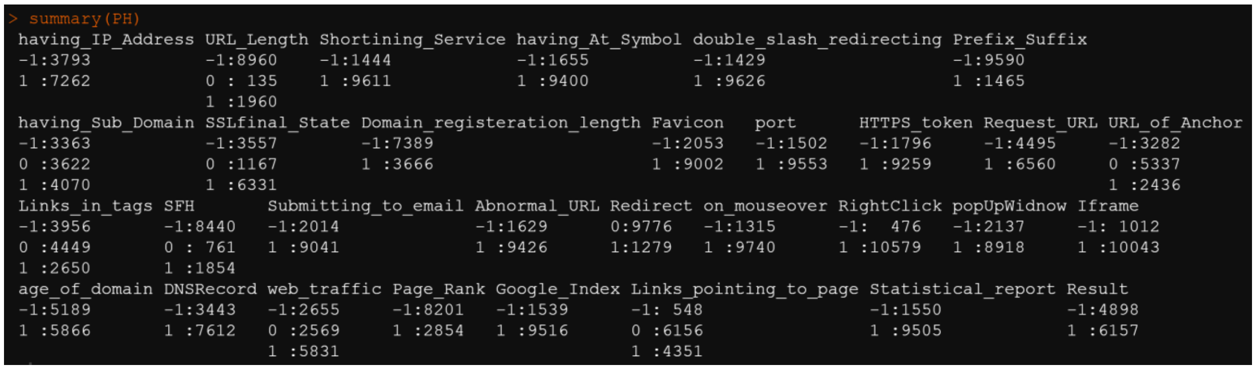
Phishing, and 0s (when present) indicating something is possibly Phishing. Not all variables contained the 0 category. While they are factors in nature, the variables were kept as integers during this analysis.

A detailed discussion of each variable and general data ruleset was documented by Mohammed et al. (n.d.). The variables are briefly discussed here in increments of 10 (since there are 30 total). Having_IP_Address (1) indicates whether an IP is present in a URL or not. This is a well-documented sign that a phishing attempt is being made. An exceptionally long URL is also indicative of a phishing attempting since phishers attempting to hide malicious websites within the length. URL_length (2) is a 3-factor variable where a URL: $<$ 54 characters is legitimate; $\geq$ 54 and $\leq$ 75 characters is suspicious; and $>$ 75 is phishing. On the other hand, URL shortening (3) is also indicative of phishing. A shortening service like TinyURL will opening a proxy website which will redirect someone to a malicious website. An example is of a shortened URL is “bit.ly/19DXSk4” (which is not malicious). Putting an @ in a URL (4) will cause the browser to disregard any URL before the @ and only direct the user to the URL after it. Another way to redirect is using “//”. A “//” will do the same as a @ symbol; however, “//” will regularly appear after HTTP and HTTPS so “//” should appropriately be in the 5th or 6th position. Thus, the rule becomes when “//” appears in the 7th or greater position, the URL is phishing (5). When the domain name contains a “-”, it puts potential victims at ease because the site looks legitimate (even though putting a “-” changes the domain completely and can lead to malicious sites)(6). A subdomain may be added to a URL. For example, http://www.hud.ac.uk/students/: “hud” is the domain, “ac” is the first subdomain denoting academic, and “uk” is the second subdomain denoting this is the United Kingdom specific website for “hud”. However, having several subdomains is uncommon and phishing links commonly use several. Thus, the rule becomes having 1 dot is legitimate, having 2 is suspicious, and having more is phishing (7). HTTPS or hyper text transfer protocol with secure sockets layer is a system used to verify that websites are legitimate. A browser can check that a website will have a certificate and that the cert is up to date before proceeding to visit the site. So, the rule becomes: if a site uses HTTPS and the issuer is trusted and the certificate is ≥ 1 year, then the site is legitimate; if a site uses HTTPS but the issuer is not trusted, then the site is suspicious; and if any other state is phishing (8). Domain registration length also serves as a proxy to know if a site is legitimate since phishing sites usually only live for a short period of time. The rule put forth is if a site domain expires in 1 year or less it is phishing; otherwise, it is legitimate (9). Many phishing websites use an externally loaded favicon, which is a graphic icon. So, if a site attempts to load a favicon from a domain which is not what domain is currently loaded, then it is phishing (10).

Ports are how information can transfer between servers. These are generally kept closed so that intruders cannot simply "walk through the front door" and take what they want. So, if a port number is of the preferred status, then it is phishing (Mohammed et al., n.d.)(11). HTTPS normally denotes that a site is safe; however, when HTTPS appears within the domain name, the website is phishing (12). "Request URL" looks at if and how much content from a site is contained within that website versus is loaded via other domains. Legitimate sites are largely self-contained. So, the rule is: a site with < 22% of request URL is legitimate; a site with ≥ 22 and < 61% of request URL is suspicious; and ≥ 61% of request URL is phishing (13). Within the JavaScript on a page, an anchor is an element with an <a> tag which has the same function as a request URL. Thus, the rule for URL of anchor is < 31% is legitimate, ≥ 31% and ≤ 67% is suspicious, and > 61% is phishing (14). Similarly, in JavaScript <Meta>, <Script>, and <Link> tags will respectively offer metadata about the HTML document, create a client side script, and retrieve other web sources. So, if there are < 17% of these links the page is legitimate; if there is ≥ 17 and ≤ 81% the page is suspicious; and > 81% is a phishing page (15). The server form handler (SFH) should almost always have the same domain name as the webpage. So, if it is empty or "about: blank", it is a phishing site (16). A web form that directs a user's input directly to an email likely is phishing. So, if a form contains "mail()" or "mailto:" commands, it is a phishing website (17). If the host name is not included in the URL ("abnormal URL"), then it is a phishing website (18). Websites that automatically redirect multiple times are phishing websites. So, if a site redirects ≤ 1 time it is legitimate; ≥ 2 and < 4 redirects is suspicious; and ≥ 4 redirects is a phishing website (19). Phishers sometimes make a fake URL appear using JavaScript when users move their cursors over the URL. So, "onMouseOver" checks if the source code has scripts to change the status bar, then it is a phishing website (20).

Similarly, phishers often will disable users from right-clicking on objects on a page to prevent them from viewing and saving a site's source code (21). Rarely do legitimate sites ask users to input personal information into a pop-up window; yet, this is a common phishing tactic (22). IFrame is an HTML tag which will show another website on a host site. It can be made very large so that no borders can be seen so that users think they are on a legitimate website. So, if the source code contains an IFrame tag, then it is a phishing website (23). As previously mentioned, most phishing sites are short lived. If the age of domain is ≥ 6 months, then it is legitimate whereas if it is < 6 months it is phishing (24). If there is no DNS record for the website's domain name or it is empty, then it is a phishing website (25). Website traffic is another proxy to help determine if a website is phishing or not since phishing sites are short lived and get little traffic. If the site ranks within the top 100,000 sites for traffic, then it is legitimate. If it is ranked higher than that, then it is suspicious. If it has no traffic or no record, then it is a phishing website (26). Page rank values range from 0 to 1 (from least to most important in terms of traffic). 95% of phishing sites have no page rank and the remaining 5% have a score of < 0.2 (27). When a site is legitimate and indexed by Google, it will appear in Google's search results. However, phishing sites are too shorted lived to be indexed. So, if they are not indexed then they are a phishing site (28). The number of links pointing to a page ground its legitimacy as well (even if the links are on the same domain). If a site has 0 referential links, then it is phishing; if a site has > 0 and ≤ 2 links, then it is suspicious; and if it has > 2 referential links, then it is legitimate (29). Lastly, multiple companies publish yearly statistical reports trending and documenting the most frequent phishing domains and IP addresses. If a site's IP or domain appears in this list, then it is a phishing site and if it does not, then it is a legitimate site (30).

Since these are all factor variables, one cannot say whether the distributions are normal, right-skewed, or left-skewed. There is no throughput pattern that all of these variables follow in their distributions into categories. The smallest category was 135 values of the "suspicious" bin (0) of URL length. The largest category was the legitimate bin (1) of Iframe with 10,043 values (Figure 4). The Result variable is distributed between legitimate sites at 55% and phishing sites at 45% (Figure 5).



```
> summary(PH)
 having_IP_Address URL_Length Shortining_Service having_At_Symbol double_slash_redirecting Prefix_Suffix
 -1:3793           -1:8960    -1:1444            -1:1655          -1:1429                  -1:9590
 1 :7262           0 : 135    1 :9611            1 :9400          1 :9626                  1 :1465
                   1 :1960
 having_Sub_Domain SSLfinal_State Domain_registeration_length Favicon   port      HTTPS_token Request_URL URL_of_Anchor
 -1:3363           -1:3557        -1:7389                     -1:2053   -1:1502   -1:1796     -1:4495     -1:3282
 0 :3622           0 :1167        1 :3666                     1 :9002   1 :9553   1 :9259     1 :6560     0 :5337
 1 :4070           1 :6331                                                                                1 :2436
 Links_in_tags SFH       Submitting_to_email Abnormal_URL Redirect on_mouseover RightClick popUpWidnow Iframe
 -1:3956       -1:8440   -1:2014             -1:1629      0:9776   -1:1315      -1:  476   -1:2137     -1: 1012
 0 :4449       0 : 761   1 :9041             1 :9426      1:1279   1 :9740      1 :10579   1 :8918     1 :10043
 1 :2650       1 :1854
 age_of_domain DNSRecord web_traffic Page_Rank Google_Index Links_pointing_to_page Statistical_report Result
 -1:5189       -1:3443   -1:2655     -1:8201   -1:1539      -1: 548                -1:1550            -1:4898
 1 :5866       1 :7612   0 :2569     1 :2854   1 :9516      0 :6156                1 :9505            1 :6157
                         1 :5831                            1 :4351
```

Figure 4. The summary command shows the distributions for the values across all 31 variables (graphic made with all variables converted to factors. All variables converted back to integers after command was run).

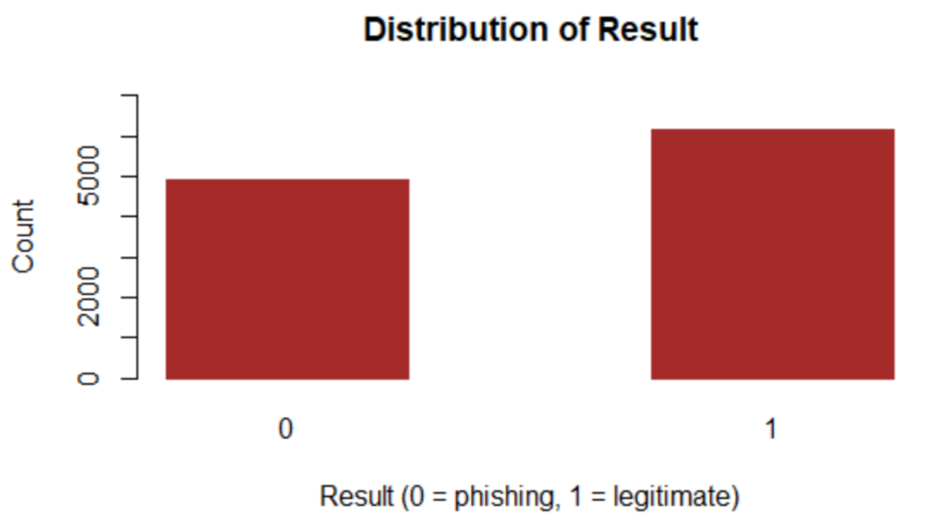


Figure 5. Distribution of the Result variable shows there were 4,898 phishing values in PH and 6,157 legitimate sites in PH.

Preprocessing:

PH required minimal data preprocessing. There were no outliers because all of the data was binary. There was no key to remove. All variables were relevant, and all rows were valuable. There were no missing values. In the Result variable, -1 was replaced with 0 for the sake of clarity.

Neural Network Method:

Neural networks are a powerful predictive method. They are described as a “black box” because the method does not describe how exactly it gets to its end result; however, they are very effective at prediction and have generally high accuracy rates (Han et al., 2011). After the user designates the dependent and independent variables, the method will observe the data. It will create a “hidden layer” with a user determined number of nodes. It is generally agreed upon that 1 layer is enough for most problems neural nets try to solve (Brownlee, 2018). Then, the program will align the independent variables, the hidden layer nodes, and the dependent variable in 3 different layers where each variable is a node. Each node is connected to every node in the layer before it and the layer after it (but is not connected to the other nodes within its layer). For example, the node of"1" from "Result" as the dependent variable will be connected to every node in the hidden layer. Also, the node of "DNSRecord" as an independent variable will be connected to every node in the hidden layer. The program will give a weight to each connection such that every node affects the nodes in front of it that it is connected to (Figure 6). The program will adjust these weights based on what it sees in the data until the weights will give the results of the dependent variable. How the program does this exactly is not shown by the algorithm and so neural networks are described as a "black box" (Han et al., 2011). Like many other predictive programs, NNs need to have a training dataset and a validation dataset. The former is used to create the algorithm while the latter is used to test the model for its accuracy.

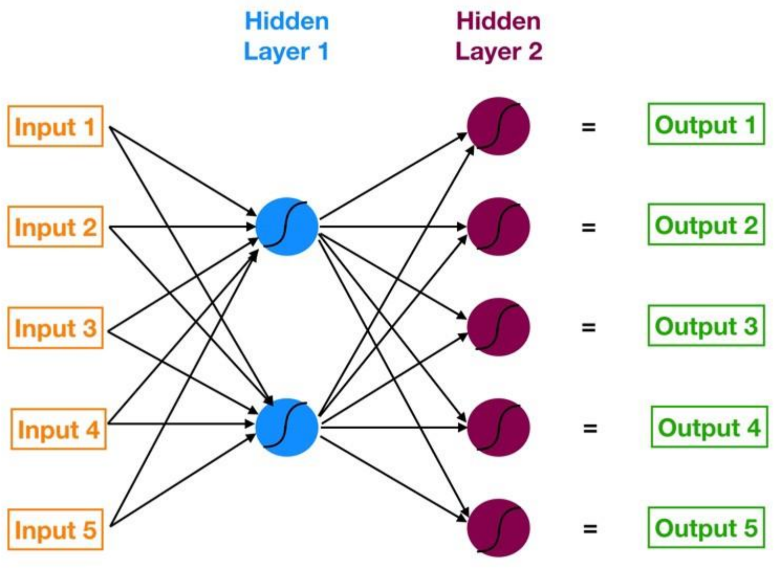
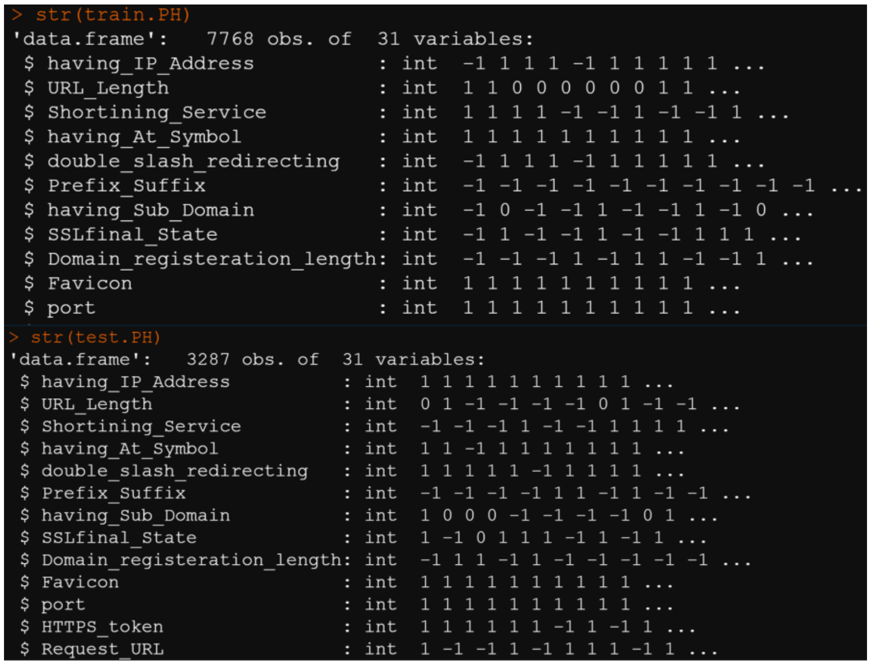


Figure 6. A neural network consists of an input layer, usually 1 hidden layer with any number of nodes, and an output layer (Yiu, 2019).

The results are made reproducible by setting a seed value at 1,234; this causes the split between the training and test datasets to occur in the same way every time (otherwise different rows would end up in the training dataset and test dataset every time this program was run). Once the split was performed, the structure command was run on both datasets to ensure it ran correctly (Figure 7). Approximately 70.3% of the rows (7,768 of 11,055) were committed to the training dataset and 29.7% to the test dataset (3,287).

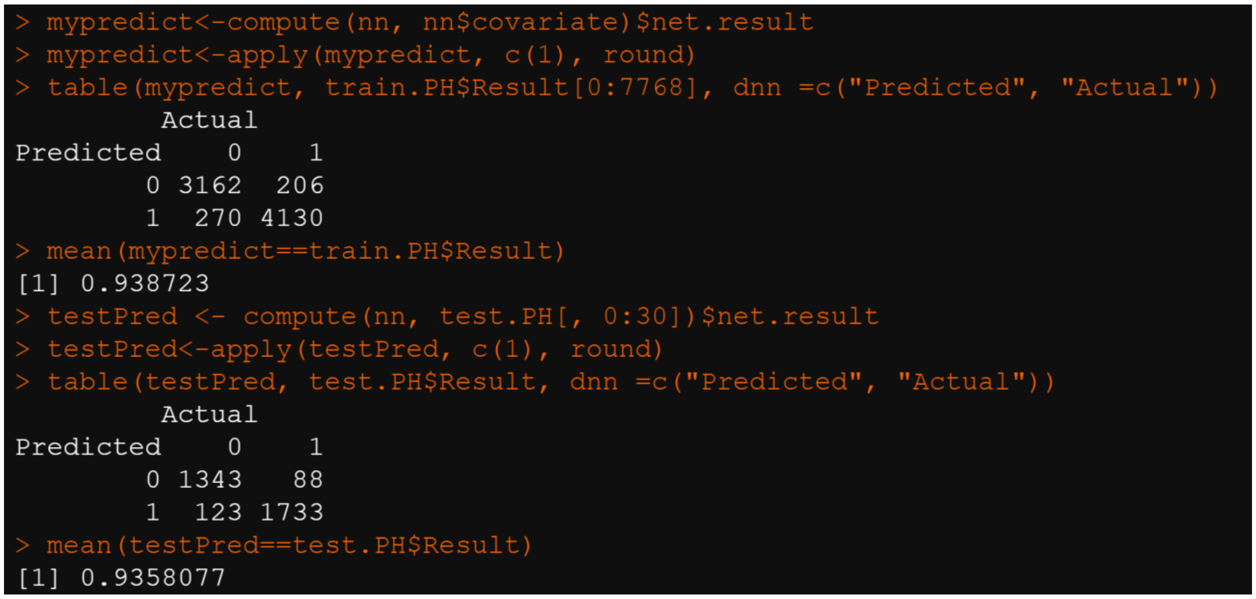


```
> str(train.PH)
'data.frame':   7768 obs. of  31 variables:
 $ having_IP_Address          : int  -1 1 1 1 -1 1 1 1 1 1 ...
 $ URL_Length                 : int  1 1 0 0 0 0 0 0 1 1 ...
 $ Shortining_Service         : int  1 1 1 1 -1 -1 1 -1 -1 1 ...
 $ having_At_Symbol           : int  1 1 1 1 1 1 1 1 1 1 ...
 $ double_slash_redirecting   : int  -1 1 1 1 -1 1 1 1 1 1 ...
 $ Prefix_Suffix              : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
 $ having_Sub_Domain          : int  -1 0 -1 -1 1 -1 -1 1 -1 0 ...
 $ SSLfinal_State             : int  -1 1 -1 -1 1 -1 -1 1 1 1 ...
 $ Domain_registeration_length: int  -1 -1 -1 1 -1 1 1 -1 -1 1 ...
 $ Favicon                    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ port                       : int  1 1 1 1 1 1 1 1 1 1 ...
> str(test.PH)
'data.frame':   3287 obs. of  31 variables:
 $ having_IP_Address          : int  1 1 1 1 1 1 1 1 1 1 ...
 $ URL_Length                 : int  0 1 -1 -1 -1 -1 0 1 -1 -1 ...
 $ Shortining_Service         : int  -1 -1 -1 1 -1 -1 1 1 1 1 ...
 $ having_At_Symbol           : int  1 1 -1 1 1 1 1 1 1 1 ...
 $ double_slash_redirecting   : int  1 1 1 1 1 -1 1 1 1 1 ...
 $ Prefix_Suffix              : int  -1 -1 -1 -1 1 1 -1 1 -1 -1 ...
 $ having_Sub_Domain          : int  1 0 0 0 -1 -1 -1 -1 0 1 ...
 $ SSLfinal_State             : int  1 -1 0 1 1 1 -1 1 -1 1 ...
 $ Domain_registeration_length: int  -1 1 1 -1 1 -1 -1 -1 -1 -1 ...
 $ Favicon                    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ port                       : int  1 1 1 1 1 1 1 1 1 1 ...
 $ HTTPS_token                : int  1 1 1 1 1 1 -1 1 -1 1 ...
 $ Request_URL                : int  1 -1 -1 1 -1 1 1 1 -1 1 ...
```

Figure 7. Structure command of the training and test dataset shows that they respectively have 7,768 rows and 3,287 rows (out of 11,055 total).

Model 1:

The first model was created by simply attempting to predict the Result variable using all 30 independent variables. For the number of hidden nodes, 2 was used. The model was then used to predict the values found in the training set. A confusion matrix was made comparing the predicted values versus the actual values of the training dataset (such that every value predicted as 0 and was found to be 0 is found in the top left corner of the matrix)(Figure 8). This yielded a 93.9% accuracy. Using the model, the values of the test dataset were then predicted with a second confusion matrix being the output. The model then yielded a 93.6% accuracy. Lastly, a graphical representation of the neural network was made (Figure 9).



```
> mypredict<-compute(nn, nn$covariate)$net.result
> mypredict<-apply(mypredict, c(1), round)
> table(mypredict, train.PH$Result[0:7768], dnn =c("Predicted", "Actual"))
         Actual
Predicted    0    1
        0 3162  206
        1  270 4130
> mean(mypredict==train.PH$Result)
[1] 0.938723
> testPred <- compute(nn, test.PH[, 0:30])$net.result
> testPred<-apply(testPred, c(1), round)
> table(testPred, test.PH$Result, dnn =c("Predicted", "Actual"))
         Actual
Predicted    0    1
        0 1343   88
        1  123 1733
> mean(testPred==test.PH$Result)
[1] 0.9358077
```

Figure 8. The first neural network model yielded a 93.9% accuracy on the training dataset and a 93.6% accuracy on the test dataset.

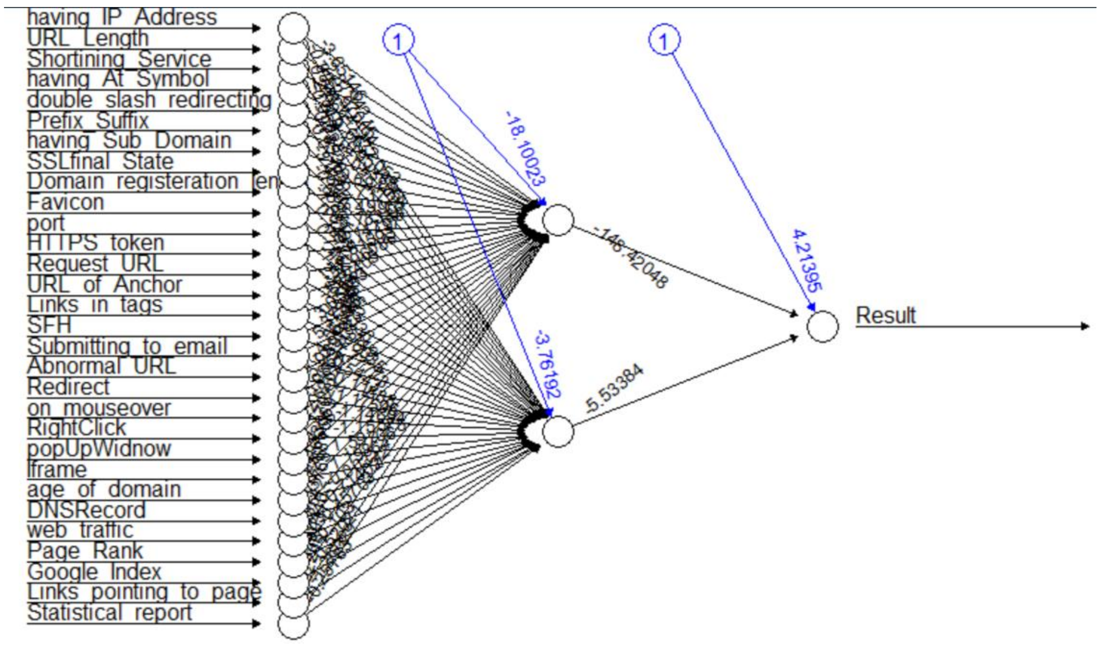
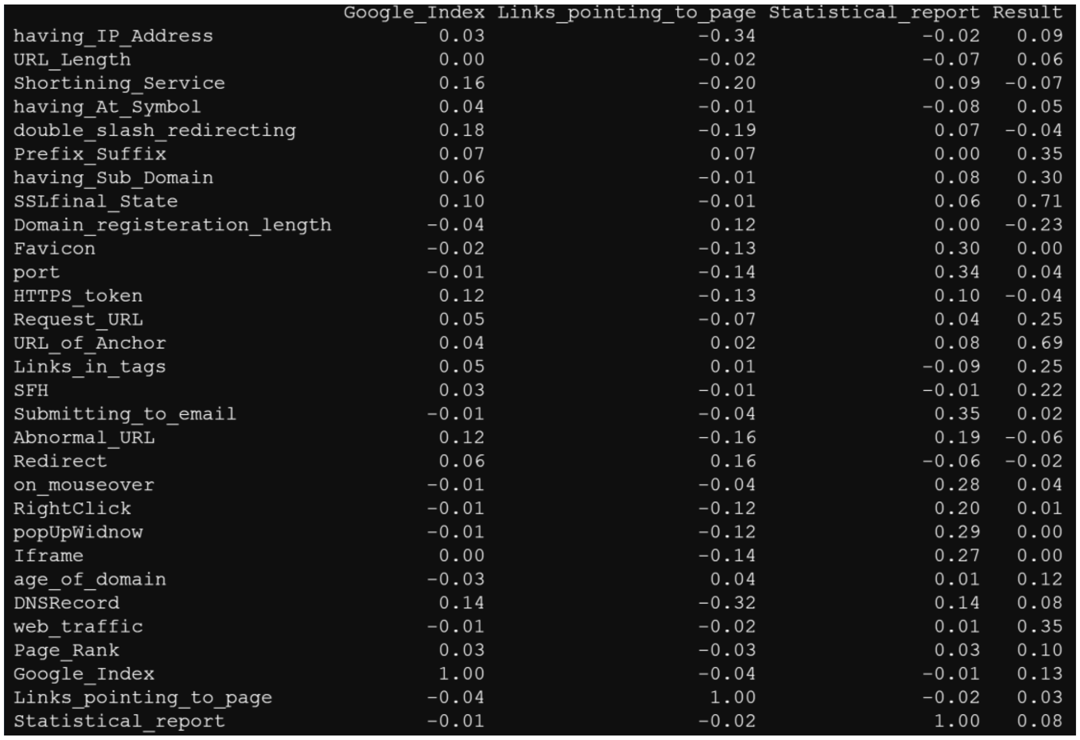


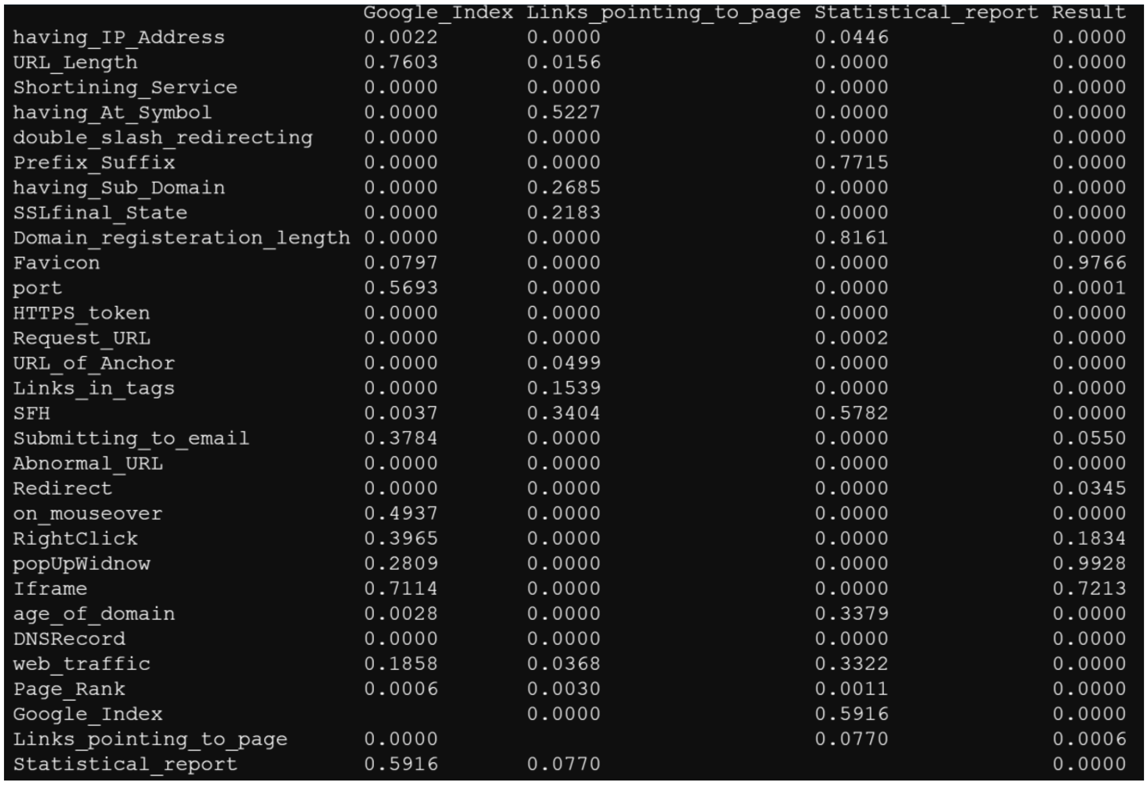
Figure 9. Graphical representation of model 1 shows the weights of each node. Not all weights can be visualized here but can be found by running the command: “nn$weights”.

Next, a correlation matrix was created in which every variable was correlated with every other variable. The output variable, the r-value, shows how closely two variables are tied together (i.e., if X variable changes 10 units, Y variable also changes 5 units). This was to see what correlations existed between the variables and the dependent variable (Result), where the data was weakest, and where the data was strongest. It was found that several variables had an r value above 0.3 with Result: Prefix suffix, having subdomain, SSL final state, URL of anchor, and web traffic (Figure 10). It was decided to make another model using just these variables for comparison. This model was named model 2. Next, the p-values were examined. A p-value indicates the likelihood the data has been observed in the configuration it did due to random chance (Han et al., 2011). So, a low p-value indicates it is unlikely to appear like that due to random chance and thus the correlation is legitimate. For most analyses, a p-value below 0.05 is deemed significant. It was found that 4 variables had a p-value of greater than 0.05 (indicating the relationships found were insignificant in nature): Iframe, popup window, submitting to email, and favicon (Figure 11). Since the dataset was not large enough for a low p-value for these categories, one cannot state definitively whether these have an impact on the Result variable or not. Thus, it was decided to make a third and final model where all variables were used except for these 4 variables.



```
                            Google_Index Links_pointing_to_page Statistical_report Result
having_IP_Address                   0.03                  -0.34              -0.02   0.09
URL_Length                          0.00                  -0.02              -0.07   0.06
Shortining_Service                  0.16                  -0.20               0.09  -0.07
having_At_Symbol                    0.04                  -0.01              -0.08   0.05
double_slash_redirecting            0.18                  -0.19               0.07  -0.04
Prefix_Suffix                       0.07                   0.07               0.00   0.35
having_Sub_Domain                   0.06                  -0.01               0.08   0.30
SSLfinal_State                      0.10                  -0.01               0.06   0.71
Domain_registeration_length        -0.04                   0.12               0.00  -0.23
Favicon                            -0.02                  -0.13               0.30   0.00
port                               -0.01                  -0.14               0.34   0.04
HTTPS_token                         0.12                  -0.13               0.10  -0.04
Request_URL                         0.05                  -0.07               0.04   0.25
URL_of_Anchor                       0.04                   0.02               0.08   0.69
Links_in_tags                       0.05                   0.01              -0.09   0.25
SFH                                 0.03                  -0.01              -0.01   0.22
Submitting_to_email                -0.01                  -0.04               0.35   0.02
Abnormal_URL                        0.12                  -0.16               0.19  -0.06
Redirect                            0.06                   0.16              -0.06  -0.02
on_mouseover                       -0.01                  -0.04               0.28   0.04
RightClick                         -0.01                  -0.12               0.20   0.01
popUpWidnow                        -0.01                  -0.12               0.29   0.00
Iframe                              0.00                  -0.14               0.27   0.00
age_of_domain                      -0.03                   0.04               0.01   0.12
DNSRecord                           0.14                  -0.32               0.14   0.08
web_traffic                        -0.01                  -0.02               0.01   0.35
Page_Rank                           0.03                  -0.03               0.03   0.10
Google_Index                        1.00                  -0.04              -0.01   0.13
Links_pointing_to_page             -0.04                   1.00              -0.02   0.03
Statistical_report                 -0.01                  -0.02               1.00   0.08
```

Figure 10. Correlation matrix shows that the variables with an r value above 0.3 are: Prefix suffix, having subdomain, SSL final state, URL of anchor, and web traffic.

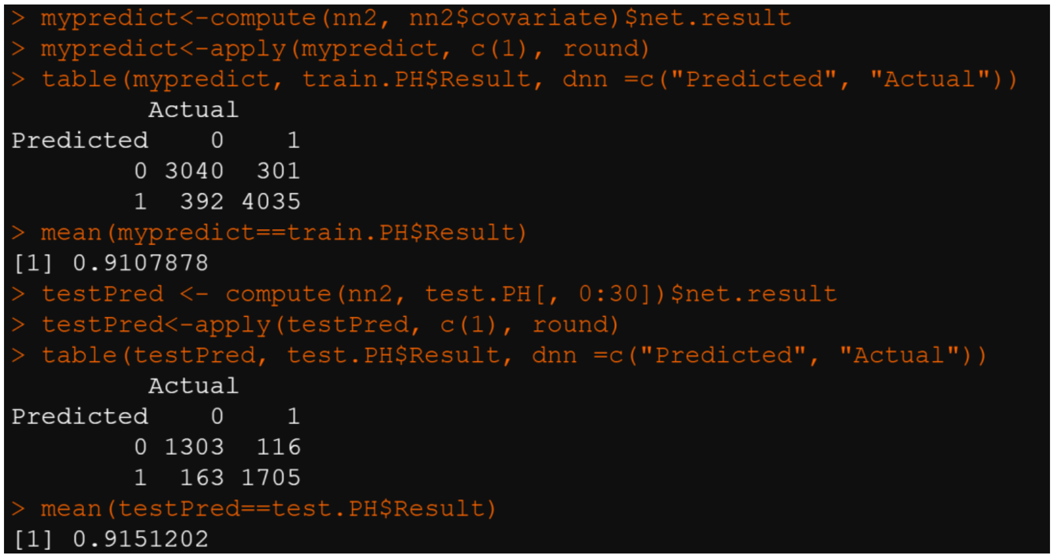


```
                            Google_Index Links_pointing_to_page Statistical_report Result
having_IP_Address           0.0022       0.0000                 0.0446             0.0000
URL_Length                  0.7603       0.0156                 0.0000             0.0000
Shortining_Service          0.0000       0.0000                 0.0000             0.0000
having_At_Symbol            0.0000       0.5227                 0.0000             0.0000
double_slash_redirecting    0.0000       0.0000                 0.0000             0.0000
Prefix_Suffix               0.0000       0.0000                 0.7715             0.0000
having_Sub_Domain           0.0000       0.2685                 0.0000             0.0000
SSLfinal_State              0.0000       0.2183                 0.0000             0.0000
Domain_registeration_length 0.0000       0.0000                 0.8161             0.0000
Favicon                     0.0797       0.0000                 0.0000             0.9766
port                        0.5693       0.0000                 0.0000             0.0001
HTTPS_token                 0.0000       0.0000                 0.0000             0.0000
Request_URL                 0.0000       0.0000                 0.0002             0.0000
URL_of_Anchor               0.0000       0.0499                 0.0000             0.0000
Links_in_tags               0.0000       0.1539                 0.0000             0.0000
SFH                         0.0037       0.3404                 0.5782             0.0000
Submitting_to_email         0.3784       0.0000                 0.0000             0.0550
Abnormal_URL                0.0000       0.0000                 0.0000             0.0000
Redirect                    0.0000       0.0000                 0.0000             0.0345
on_mouseover                0.4937       0.0000                 0.0000             0.0000
RightClick                  0.3965       0.0000                 0.0000             0.1834
popUpWidnow                 0.2809       0.0000                 0.0000             0.9928
Iframe                      0.7114       0.0000                 0.0000             0.7213
age_of_domain               0.0028       0.0000                 0.3379             0.0000
DNSRecord                   0.0000       0.0000                 0.0000             0.0000
web_traffic                 0.1858       0.0368                 0.3322             0.0000
Page_Rank                   0.0006       0.0030                 0.0011             0.0000
Google_Index                             0.0000                 0.5916             0.0000
Links_pointing_to_page      0.0000                              0.0770             0.0006
Statistical_report          0.5916       0.0770                                    0.0000
```

Figure 11. Correlation matrix shows that the variables with a p value above 0.05 (non-significant values) are: Iframe, popup window, submitting to email, and favicon

Model 2:

The same steps were taken to make the second as were taken to form the first model. The model was made using 2 hidden layer nodes (with only 1 hidden layer). As previously mentioned, 5 variables used: Prefix suffix, having subdomain, SSL final state, URL of anchor, and web traffic. Once the model was created using the training dataset, it was tested both on the training dataset and the test dataset. The confusion matrices below (Figure 12) show the raw prediction numbers (both the correctly predicted values and the failures). The training dataset had an accuracy of 91.1% and the test dataset had an accuracy of 91.5%. A graphical representation of the network was also generated (Figure 13).



```
> mypredict<-compute(nn2, nn2$covariate)$net.result
> mypredict<-apply(mypredict, c(1), round)
> table(mypredict, train.PH$Result, dnn =c("Predicted", "Actual"))
         Actual
Predicted    0    1
        0 3040  301
        1  392 4035
> mean(mypredict==train.PH$Result)
[1] 0.9107878
> testPred <- compute(nn2, test.PH[, 0:30])$net.result
> testPred<-apply(testPred, c(1), round)
> table(testPred, test.PH$Result, dnn =c("Predicted", "Actual"))
         Actual
Predicted    0    1
        0 1303  116
        1  163 1705
> mean(testPred==test.PH$Result)
[1] 0.9151202
```

Figure 12. The second neural network model yielded a 91.1% accuracy on the training dataset and a 91.5% accuracy on the test dataset.

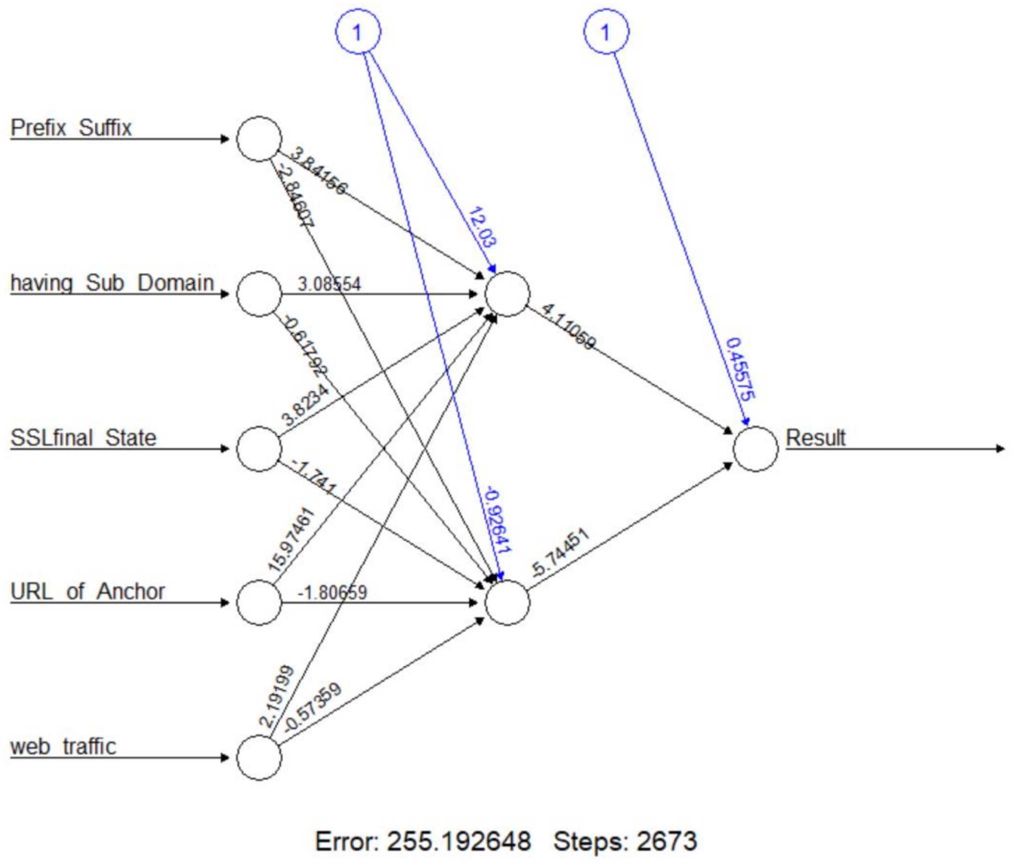
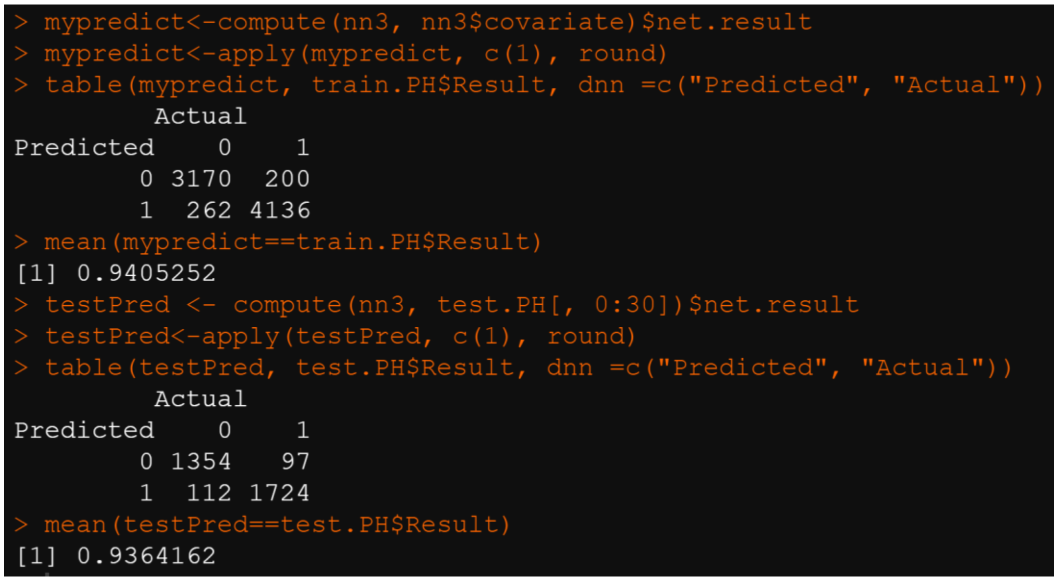


Figure 13. Graphical representation of model 2 shows the weights of each node. Not all weights can be visualized here but can be found by running the command: "nn$weights".

Model 3:

Finally, model 3 was created using the same steps as the previous models but removed 4 variables because they had non-significant p-values: Iframe, popup window, submitting to email, and favicon. Once the model was generated using all the other independent variables, it was tested for accuracy against the training dataset and the test dataset. The confusion matrices showed the raw values for correctly and incorrectly predicted values (Figure 14). The third model was found to have a 94.1% accuracy on the training dataset and a 93.6% accuracy on the test dataset.



```
> mypredict<-compute(nn3, nn3$covariate)$net.result
> mypredict<-apply(mypredict, c(1), round)
> table(mypredict, train.PH$Result, dnn =c("Predicted", "Actual"))
         Actual
Predicted    0    1
        0 3170  200
        1  262 4136
> mean(mypredict==train.PH$Result)
[1] 0.9405252
> testPred <- compute(nn3, test.PH[, 0:30])$net.result
> testPred<-apply(testPred, c(1), round)
> table(testPred, test.PH$Result, dnn =c("Predicted", "Actual"))
         Actual
Predicted    0    1
        0 1354   97
        1  112 1724
> mean(testPred==test.PH$Result)
[1] 0.9364162
```

Figure 14. The third neural network model yielded a 94.1% accuracy on the training dataset and a 93.6% accuracy on the test dataset.

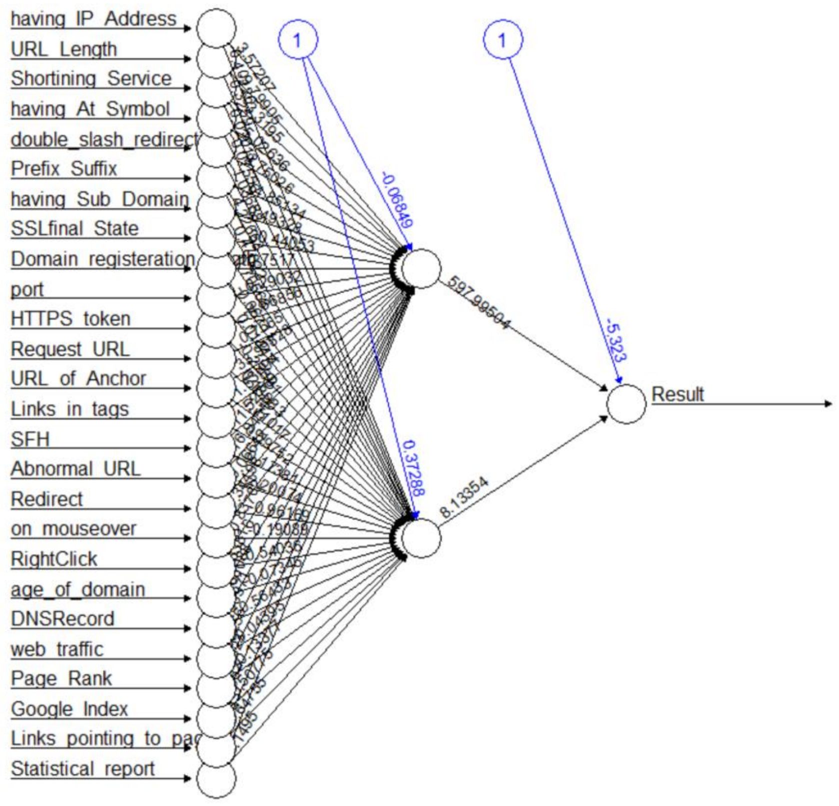


Figure 15. Graphical representation of model 3 shows the weights of each node. Not all weights can be visualized here but can be found by running the command: “nn$weights”.

Results and Model Evaluation:

Thus, 3 models were made. The first used all 30 independent variables to predict the dependent variable, Result. The second used 5 variables since they had an r-value above 0.3 (and significant p-values): Prefix suffix, having subdomain, SSL final state, URL of anchor, and web traffic. Lastly, the third model used 26 of the independent variables to predict results since 4 variables had insignificant p-values in their correlations with Result: Iframe, popup window, submitting to email, and favicon. The accuracy results were summarized in Table 1.

Model	Training Acc.	Test Acc.	Avg. Acc.
1	93.9	93.6	93.8
2	91.1	91.5	91.3
3	94.1	93.6	93.9

Table 1. The accuracy (acc.) results show that models 1 and 3 had relatively equal results while model 2 was about 2.5% less accurate.

It is clear from the results that models 1 and 3 had the most accurate results being nearly equivalent (there was only a 0.1% different between their respective averages). On the other hand, model 2 was 2.5% less accurate. This is not a terrible accuracy difference; however, it is an exceptional score when considering that this model was made by only 5 variables. Thus, a majority of the results were found in just 16.67% of the variables. In a since, this dataset then follows a pareto distribution where 80% of the results are found in 20% (or less) of the results (Tanabe, 2018). While the other two models were more accurate, this model is much leaner. Using only 5 variables means evaluating 55,275 datapoints (5 x 11,055 = 55,275) instead of 331,650 (30 x 11,055 = 331,650). This can save a significant amount of time.

Thus, it depends on the needs of the programmers using these types of models. If their goal is speed, then they should use models like model 2 analyzing only a portion of the most valuable variables. But if their goal is to have as high of an accuracy as possible (regardless of how long it takes to analyze), then they should aim for models like 1 and 3. Alternatively, they could use a combination of both of these sorts of models. As messages are received, they could enter a quarantine zone before being available to the user. The faster model (2) would analyze it first. If the algorithm could not identify it with a high level of certainty quickly, then it would hold onto it in the quarantine it further and run it through a more accurate, slower model (like 1/3). Since messaging demands speediness at near instant speeds nowadays, developers cannot allow messages to get caught in quarantine zones too long. Also, the vast majority of messages one might receive are going to be non-phishing. Fortunately, developers can use human intelligence as a fallback. Most messaging apps now give users access to the Spam folder so that users can check it for false negatives. The apps typically also have a button to mark messages as phishing in the case of false positives (which in this case helps train the algorithms better for future predictions).

It is fascinating that 4 variables were found to have no significant relationship with the Result. This is surprising since these variables were specifically added to this dataset because they are supposed to be able to help predict when messages are phishing (Mohammed et al., n.d.). It is possible that these normally are great indicators of phishing attempts, but the dataset did not show it due to random chance. This is not very likely because the dataset is quite large. However, it is critical to note that each of these 4 variables are not absolutely indicative of phishing attempts but only possible indicators of phishing. All of these may be used by legitimate websites (even if they may not be best practice). In contrast, Mohammed et al. (n.d.) note that if an IP address appears in a URL (or the hexadecimal equivalent), then this is a sure sign that the website is a phishing attempt.

On the other end of the spectrum, it is interesting to look at the 5 variables which had a significant r-value greater than 0.3. In an attempt to discover a driving measure behind them, all the variables with "suspicious" (denoted as 0) were examined (Table 2). Four of the variables had an r-value greater than 0.3 and were used to build model 2 (having_Sub_Domain, SSLfinal_State, URL_of_Anchor, and web_traffic). Two of the variables (Links_in_tags and SFH) had decently sized r-values but two had very small r-values (URL_Length and Links_pointing_to_page). It is critical to note that all 8 of these variables had significant p-values. There was also no standout pattern noticed when examining the size of the suspicious category or ratios between suspicious, phishing, and legitimate either (for all sizes see Figure 3). The only pattern observed is that the 4 variables used for model 2 all had over 1,000 values in the suspicious category. However, this does not provide an explanatory pattern for why those 5 variables had significant r-values.

Variable	**Number of Suspicious**	**r-value**	**Comment**
URL Length	135	0.06	N/A
having_Sub_Domain	3633	0.3	Model 2
SSLfinal_State	1167	0.71	Model 2
URL_of_Anchor	5337	0.69	Model 2
Links_in_tags	4449	0.25	N/A
SFH	761	0.22	N/A
web_traffic	2569	0.35	Model 2
Links_pointing_to_page	6156	0.03	N/A

Table 2. Of the 8 variables with "suspicious" values (0), 4 had r-values greater than 0.3 and 4 did not.

Regardless, it has been shown that a neural network can predict whether a message is phishing or not with a 93-94% accuracy. Also, it was shown that this can be predicted with 91% accuracy using only 5 of the variables showing that this data follows a pareto distribution (in the sense that over 80% of the results are contained within less than 20% of the data).

Conclusion:

It is critical to have software to predict incoming phishing attempts. With the sharp increase in phishing attacks, companies must work hard to safeguard themselves and their data. Companies should educate their employees to be savvy: do not open emails if you do not know who they are from; hover over links to examine the URL before opening it; and looking out for subtle hallmarks of impersonation (like improper logos, crude spelling mistakes in messages or URLs, etc.). The cost of not preventing data breaches is far too high. These sorts of features are critical when it comes to spam/phishing identification. The demands of users are steep: messaging must be fluid and rapid while preventing phishing attempts. The anti-phishing programs are always working in the background to keep users safe. It has thus been shown that a simple program can be made using minimal tuning but having a high accuracy (93-94%). If a program can prevent about 94% of phishing attempts automatically, with a simple program, they can prevent phishing at much higher rates using more complicated safety nets. While there is a tradeoff between speed and safety, a company's cybersecurity needs to focus on making sure that users will be happy with seamless messaging and that companies will stay safer, be more productive, and protect their assets.

Limitations and Improvements:

There are a few areas which could still be explored. First, it would be interesting to iterate a series of new models increasing the number of nodes from 2 to 6. The general rule of nodes is that there should be a number between the dependent variable and the independent variables (Brownlee, 2018). This is another critical parameter which could be used to fine-tune this model. Second, it would be interesting to make another model where it only used independent variables with an r-value of 0.1 or greater (and having p-values that are significant). While neural nets are extremely accurate, their downfall is that they tend to overfit the data. So, it would be interesting to cut out any variables which had very little correlation with Result. It would be expected that the model would not overfit and produce better results than model 1 or 3. Lastly, it would be interesting to perform further research into why 80% of the results derived from just 5 variables. Was that pattern just due to this dataset? Was it because those are commonly used techniques by hackers? If so, why are those tools particularly effective? These questions have answers, but will need to be tackled another time.

References:

APWG. (2021). Phishing Activity Trends Report 4th Quarter 2020. APWG. https://docs.apwg.org/reports/apwg_trends_report_q4_2020.pdf

Brownlee, J. (2018). How to Configure the Number of Layers and Nodes in a Neural Network. Machine Learning Mastery. Retrieved from: https://machinelearningmastery.com/how-to-configure-the-number-of-layers-and-nodes-in-a-neural-network/

Han, J., Kamber, M., and Pei, J. (2011). Data Mining: Concepts and Techniques, Third Edition. Elsevier. Retrieved June 6th, 2021 from:

http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf

IBM Security. (2021). Cost of a Data Breach Report. IBM. Retrieved from: https://www.ibm.com/downloads/cas/RZAX14GX

Mohammed, R., Thabtah, F., McCluskey, L., (n.d.). Phishing Websites Features. UCI. Retrieved from: http://archive.ics.uci.edu/ml/machine-learning-databases/00327/

Rosenthal, M. (2021). Must-Know Phishing Statistics: Updated 2021. Tessian. Retrieved from: https://www.tessian.com/blog/phishing-statistics-2020/

Sommestad, T., & Karlzén, H. (2019). A meta-analysis of field experiments on phishing susceptibility. In 2019 APWG symposium on electronic crime research (eCrime) (pp. 1-14). IEEE.

Tanabe, K. (2018). Pareto's 80/20 rule and the Gaussian distribution. *Physica A: Statistical Mechanics and its Applications*, *510*, 635-640.

UCI Machine Learning Lab (2015). Phishing Websites Data Set. UCI. Retrieved from: http://archive.ics.uci.edu/ml/datasets/Phishing+Websites#

Yiu, J. (2019). Understanding Neural Networks. Towards Data Science. Retrieved from: https://towardsdatascience.com/understanding-neural-networks-19020b758230