400 Capstone Project

Stephen Ramotowski
sjramotowski@wm.edu
Computer Science Department
William & Mary
Williamsburg, Virginia, USA

ABSTRACT

This project tackles the problem of network traffic classification by using two simple models to determine their effectiveness. The goal of the project was to use a model to correctly classify the source website of a thirty-second-long network data stream. I wanted to test whether two simple models I learned in an introductory data science class would be able to successfully identify the network data and contribute to the field of network traffic classification.

The two models created were a k-Nearest Neighbor model and a logistic regression model. Testing of the models showed that the k-NN model had a higher accuracy (0.8) than logistic regression (0.49). The validation of the models showed that the k-NN model was able to identify the source website correctly for all four data streams while the logistic regression model only identifies one correctly.

This project demonstrates the effectiveness of two different simple models and shows the feasibility of using them in the field of network traffic classification where network security, monitoring, and management are important [2].

1 INTRODUCTION

The goal of this project is to identify a website solely based on its network traffic. I collected data from four popular websites: Google [8], Weather.com [5], Wikipedia [3], and Blackboard [1].

To classify the websites, I created two models, a k-Nearest Neighbor model, and a logistic regression model. To create the k-NN model, k-fold cross-validation was used to find the optimal value of k. These models classify individual packets of network data and use a majority to classify the source website of the stream of network data. The accuracy of classifying individual packets of the k-NN model (0.8) was significantly higher than the accuracy of the logistic regression (0.49). This is reflected in the ability of the models to classify the source website where the k-NN model successfully identified all four websites while the logistic regression only identified one.

This project also includes a list of future improvements to implement. This project contributes to the field of network traffic classification and demonstrates the effectiveness of two different models.

2 PROPOSED METHOD

This section presents my methodology for monitoring, analyzing, and classifying computer network traffic based on its originating website. This section explores the data collection method, the features chosen, the two models created, and how they work.

2.1 Data Preparation

I used Wireshark [4], a network traffic and packet analyzer to collect the data for this project. Wireshark is a free and open-source tool used by millions of people around the world. Wireshark captures the packets of network traffic, including the size and protocol used. Wireshark can filter traffic by source website allowing data collection for specific websites. Wireshark has an intuitive graphical interface that makes exploring and collecting network traffic a simple task. These features are integral to the classification task of the project.

I could easily collect network traffic data for all four websites at the packet level using Wireshark. Wireshark allows data to be extracted as a CSV file which Microsoft Excel supports. Excel is a powerful spreadsheet software program that offers data visualization and analysis tools. Using Excel I could calculate new features for my models to use.

2.2 Data Collection

I initiated the data collection process by opening Wireshark. For each website, I created a capture filter for

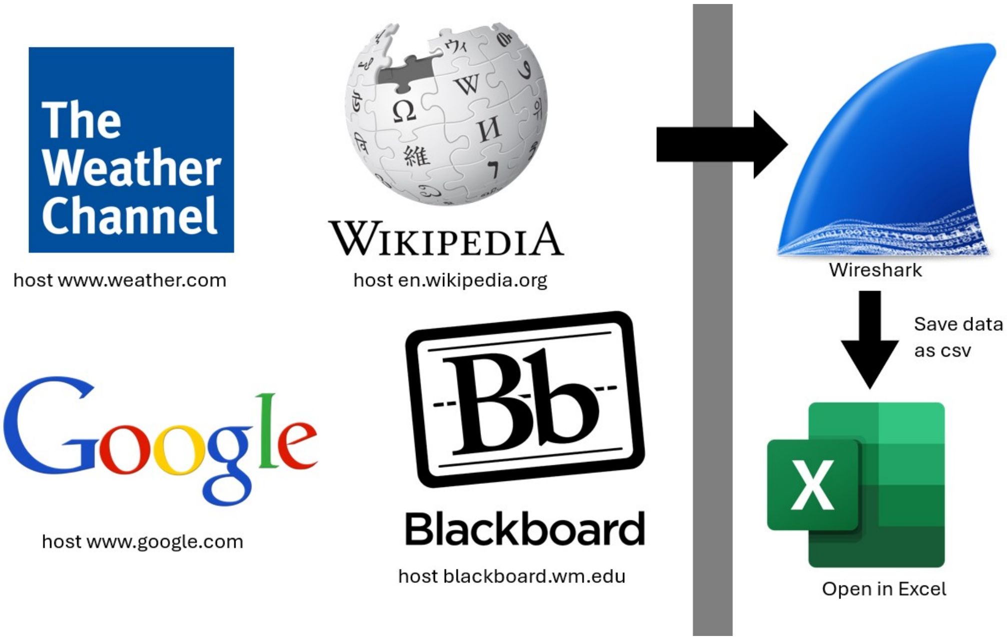
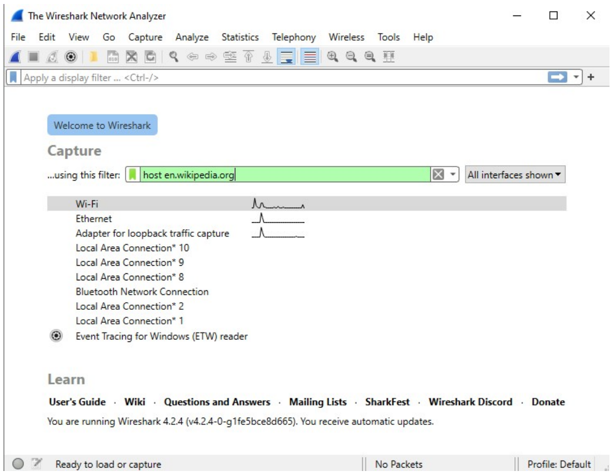


Figure 1: Process of Collecting Data


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate

...using this filter: host en.wikipedia.org

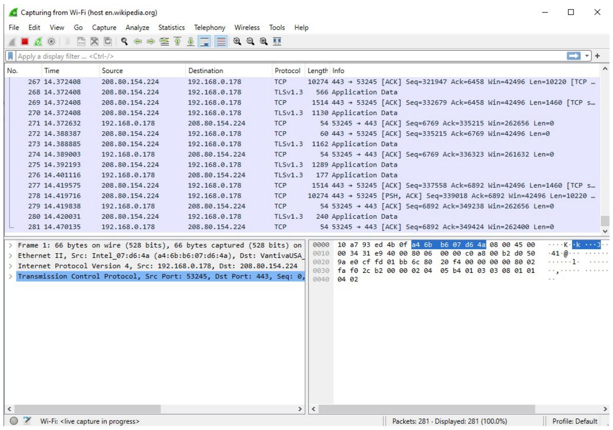
All interfaces shown

Wi-Fi
Ethernet
Adapter for loopback traffic capture
Local Area Connection* 10
Local Area Connection* 9
Local Area Connection* 8
Bluetooth Network Connection
Local Area Connection* 2
Local Area Connection* 1
Event Tracing for Windows (ETW) reader

You are running Wireshark 4.2.4 (v4.2.4-0-g1fe5bce8d665). You receive automatic updates.

Ready to load or capture | No Packets | Profile: Default

Figure 2: Wireshark Interface


Capturing from Wi-Fi (host en.wikipedia.org)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

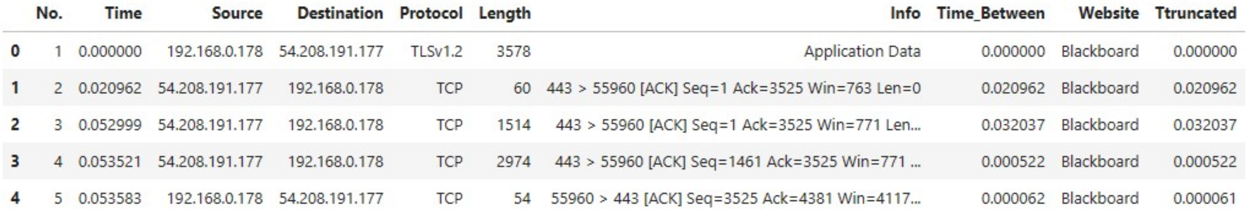
```
No.   Time       Source          Destination     Protocol Length Info
  267 14.372408  208.80.154.224  192.168.0.178   TCP      10274 443 → 53245 [ACK] Seq=321947 Ack=6458 Win=42496 Len=10220 [TCP …
  268 14.372408  208.80.154.224  192.168.0.178   TLSv1.3    566 Application Data
  269 14.372408  208.80.154.224  192.168.0.178   TCP       1514 443 → 53245 [ACK] Seq=332679 Ack=6458 Win=42496 Len=1460 [TCP s…
  270 14.372408  208.80.154.224  192.168.0.178   TLSv1.3   1130 Application Data
  271 14.372632  192.168.0.178   208.80.154.224  TCP         54 53245 → 443 [ACK] Seq=6769 Ack=335215 Win=262656 Len=0
  272 14.388387  208.80.154.224  192.168.0.178   TCP         60 443 → 53245 [ACK] Seq=335215 Ack=6769 Win=42496 Len=0
  273 14.388885  208.80.154.224  192.168.0.178   TLSv1.3   1162 Application Data
  274 14.389003  192.168.0.178   208.80.154.224  TCP         54 53245 → 443 [ACK] Seq=6769 Ack=336323 Win=261632 Len=0
  275 14.392193  208.80.154.224  192.168.0.178   TLSv1.3   1289 Application Data
  276 14.401116  192.168.0.178   208.80.154.224  TLSv1.3    177 Application Data
  277 14.419575  208.80.154.224  192.168.0.178   TCP       1514 443 → 53245 [ACK] Seq=337558 Ack=6892 Win=42496 Len=1460 [TCP s…
  278 14.419716  208.80.154.224  192.168.0.178   TCP      10274 443 → 53245 [PSH, ACK] Seq=339018 Ack=6892 Win=42496 Len=10220 …
  279 14.419838  192.168.0.178   208.80.154.224  TCP         54 53245 → 443 [ACK] Seq=6892 Ack=349238 Win=262656 Len=0
  280 14.420031  208.80.154.224  192.168.0.178   TLSv1.3    240 Application Data
  281 14.470135  192.168.0.178   208.80.154.224  TCP         54 53245 → 443 [ACK] Seq=6892 Ack=349424 Win=262400 Len=0

> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on
> Ethernet II, Src: Intel_07:d6:4a (a4:6b:b6:07:d6:4a), Dst: VantivaUSA_
> Internet Protocol Version 4, Src: 192.168.0.178, Dst: 208.80.154.224
> Transmission Control Protocol, Src Port: 53245, Dst Port: 443, Seq: 0,

0000  10 a7 93 ed 4b 0f a4 6b  b6 07 d6 4a 08 00 45 00   ····K·k ···J··
0010  00 34 31 e9 40 00 80 06  00 00 c0 a8 00 b2 d0 50   ·41·@··· ······
0020  9a e0 cf fd 01 bb 6c 80  20 f4 00 00 00 00 80 02   ······l· ·····
0030  fa f0 2c b2 00 00 02 04  05 b4 01 03 03 08 01 01   ··,····· ······
0040  04 02                                              ··
```

Wi-Fi: <live capture in progress> | Packets: 281 · Displayed: 281 (100.0%) | Profile: Default

Figure 3: Wireshark Collecting Network Data

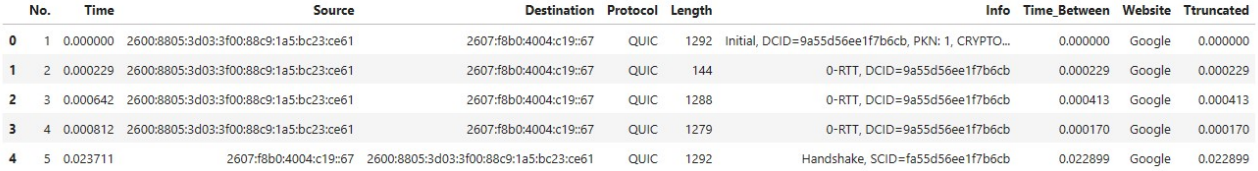
the Wi-Fi connection. First, I would apply the filter, and then I would visit the website and navigate its pages. After enough time or packets collected browsing the website, I would stop the collection by Wireshark and export the data as a CSV file. I repeated this for each website: Google, Weather.com, Blackboard, and Wikipedia. This resulted in four CSV files containing the raw network traffic data. This data contained many details such as Time, Source, Destination, Information, Protocol, and Length. I also used this same method to collect the validation data which was thirty seconds of network traffic for each website. Figure 2 shows the initial interface of Wireshark where a capture filter is set. Figure 3 shows the interface where the data collection takes place.

After collecting raw data using Wireshark, I would open each CSV file in Excel to add features. Using Excel I added a column, Website, that contained the source website of the data. This would be the target for my models. I also used an Excel formula to calculate the time between packets. The formula subtracted the time



	No.	Time	Source	Destination	Protocol	Length	Info	Time_Between	Website	Ttruncated
0	1	0.000000	192.168.0.178	54.208.191.177	TLSv1.2	3578	Application Data	0.000000	Blackboard	0.000000
1	2	0.020962	54.208.191.177	192.168.0.178	TCP	60	443 > 55960 [ACK] Seq=1 Ack=3525 Win=763 Len=0	0.020962	Blackboard	0.020962
2	3	0.052999	54.208.191.177	192.168.0.178	TCP	1514	443 > 55960 [ACK] Seq=1 Ack=3525 Win=771 Len...	0.032037	Blackboard	0.032037
3	4	0.053521	54.208.191.177	192.168.0.178	TCP	2974	443 > 55960 [ACK] Seq=1461 Ack=3525 Win=771 ...	0.000522	Blackboard	0.000522
4	5	0.053583	192.168.0.178	54.208.191.177	TCP	54	55960 > 443 [ACK] Seq=3525 Ack=4381 Win=4117...	0.000062	Blackboard	0.000061

Figure 4: Table Showing Features from Blackboard data Before Dropping

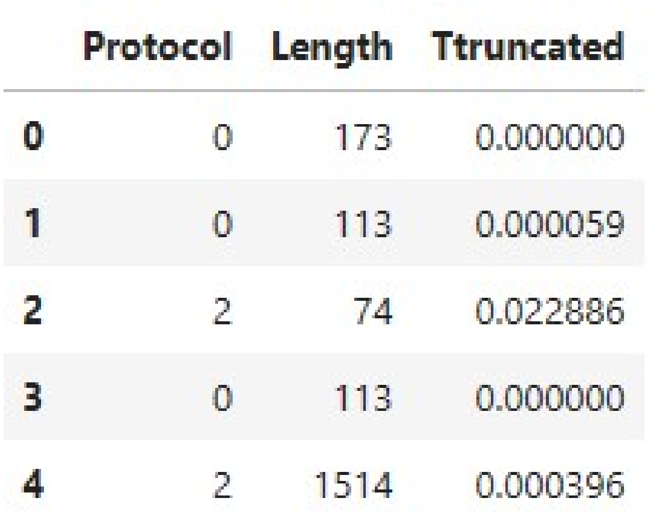


	No.	Time	Source	Destination	Protocol	Length	Info	Time_Between	Website	Ttruncated
0	1	0.000000	2600:8805:3d03:3f00:88c9:1a5:bc23:ce61	2607:f8b0:4004:c19::67	QUIC	1292	Initial, DCID=9a55d56ee1f7b6cb, PKN: 1, CRYPTO...	0.000000	Google	0.000000
1	2	0.000229	2600:8805:3d03:3f00:88c9:1a5:bc23:ce61	2607:f8b0:4004:c19::67	QUIC	144	0-RTT, DCID=9a55d56ee1f7b6cb	0.000229	Google	0.000229
2	3	0.000642	2600:8805:3d03:3f00:88c9:1a5:bc23:ce61	2607:f8b0:4004:c19::67	QUIC	1288	0-RTT, DCID=9a55d56ee1f7b6cb	0.000413	Google	0.000413
3	4	0.000812	2600:8805:3d03:3f00:88c9:1a5:bc23:ce61	2607:f8b0:4004:c19::67	QUIC	1279	0-RTT, DCID=9a55d56ee1f7b6cb	0.000170	Google	0.000170
4	5	0.023711	2607:f8b0:4004:c19::67	2600:8805:3d03:3f00:88c9:1a5:bc23:ce61	QUIC	1292	Handshake, SCID=fa55d56ee1f7b6cb	0.022899	Google	0.022899

Figure 5: Table Showing Features from Google data Before Dropping

of the packet by the time of the previous packet which results in the time between the two. Finally, I used another Excel formula to truncate the calculated time between values to remove the scientific notation that Excel automatically applies. The result of processing the data is three new columns: Time_Between, Website, and Ttruncated.

The next step in my process was to combine the four separate files of network traffic data into one data frame and drop the unwanted features. The three features I extracted from the raw data and used in modeling were Length, Protocol, and Ttruncated. Figures 4 and 5 shows an example of the data before unwanted features are dropped. Figure 6 shows the features used to create the model.



	Protocol	Length	Ttruncated
0	0	173	0.000000
1	0	113	0.000059
2	2	74	0.022886
3	0	113	0.000000
4	2	1514	0.000396

Figure 6: Table Showing Features without Target

2.3 Length

The length of a packet determines the size of the whole packet including its header, data, and trailer. Different types of packets have varying lengths depending on the data volume transferred. The length can offer insight into the content of a website. For example, a website that contains mostly text will on average have smaller packets than a website that streams media or contains images.

2.4 Protocol

The protocol of a packet determines the format of the packets and how they are sent. Websites use different protocols for their packets. For example, one website might use only TCP as a protocol, and another website might only use UDP. This makes protocol a useful feature for classifying the network traffic of a website.

2.5 Ttruncated

This feature is the time between packets truncated to six decimal points. A website that sends more packets will on average have less time between packets than a website that sends fewer packets. Since the goal of this project is to classify thirty seconds of network traffic as a specific website, this feature is useful for representing the number of packets sent over time.

3 MODELING

To classify the source website of network traffic, I created and trained two models. The first model used logistic regression and was a baseline to compare with the second model which used a K-Nearest Neighbor algorithm (k-NN).

3.1 Logistic Regression

Logistic regression was chosen as a baseline model because of its simplicity. Logistic regression is a supervised learning method used for classification. Logistic regression predicts probability values using a logistic function [6].

3.2 What is k-NN

k-NN is A supervised learning method used for classification and regression. An object is classified based on a plurality vote of its neighbors, with k representing the number of neighbors. k-NN is a simple model that is easy to implement, highly adaptable, and only has one hyperparameter [7].

- If k is equal to one, then an object is classified as the class of its nearest neighbor.
- In Figure 7, if k is equal to 3 (represented by the orange ring), the yellow point is classified as a red square.
- If k is equal to 7 (represented by the green ring), the yellow point is classified as a blue triangle.

The number of neighbors has a huge impact on the classification of an object.

3.3 Employed Method

After collecting all the data from each website into a single data frame and dropping the unwanted features, I began the process of selecting the optimal value of k, the number of neighbors. I used k-fold cross-validation which creates and averages the scores of multiple models using different test-train splits for each value of k. The optimal value of k is the value that had the highest mean testing score, or the score for data the model had not seen or trained with. I tested values of k between two and one hundred. The optimal value of k was determined to be thirty-eight which is shown in figure 8. Using this value of k, I created a single test-train split and trained a k-NN model.

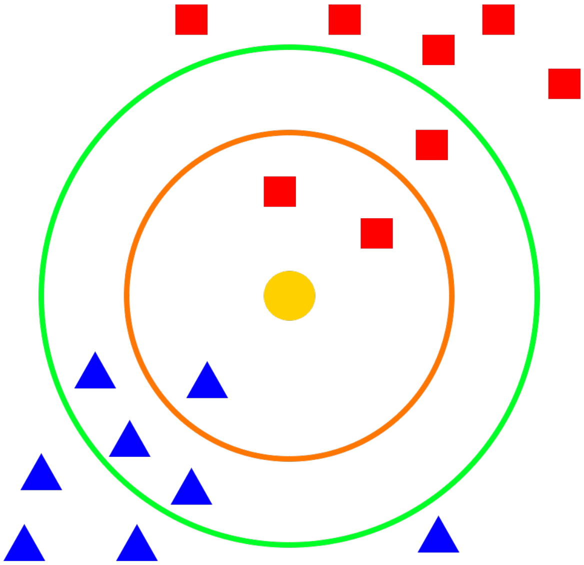


Figure 7: Example image for k-NN

4 EVALUATION AND RESULTS

As shown as in figure 9, The test accuracy of the k-NN model was 0.804 and the test accuracy of the logistic regression model was 0.493. These results represent the accuracy of identifying a single packet correctly. The goal of this project is to identify the source website of a thirty-second stream of network data. By inputting the validation data discussed previously I was able to see if the models were able to identify the source website correctly. A model correctly identifies a website if it classifies a majority of packets as the correct website. The results of this validation show that the k-NN model was able to identify the website correctly for the four validation data sets while the logistic regression was only able to identify Google correctly. These findings are shown in figure 12, which has the accuracy for each validation

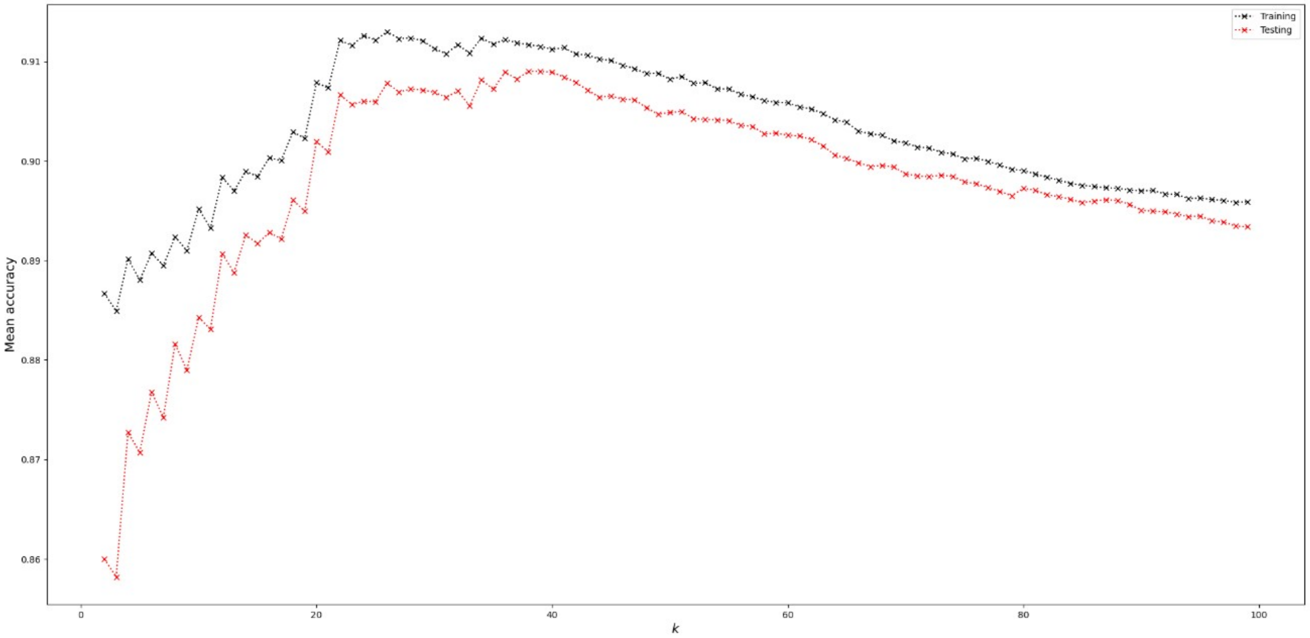
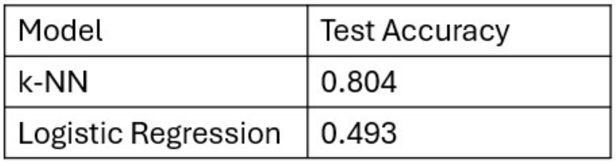
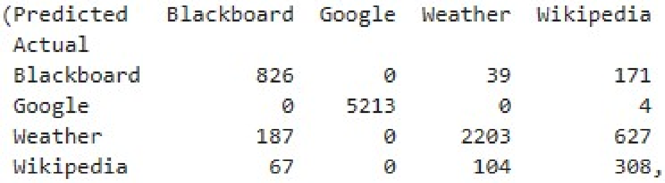


Figure 8: Graph of mean test and training scores from k-fold cross-validation. Optimal k is 38



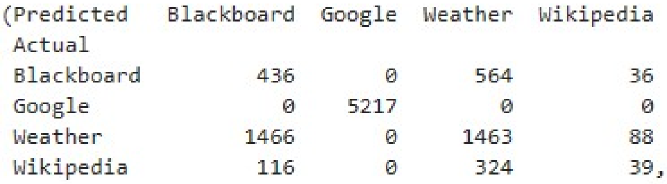
Model	Test Accuracy
k-NN	0.804
Logistic Regression	0.493

Figure 9: Table Showing Accuracy of Each Model Classifying Individual Packets



```
(Predicted   Blackboard  Google  Weather  Wikipedia
 Actual
 Blackboard         826       0       39        171
 Google               0    5213        0          4
 Weather            187       0     2203        627
 Wikipedia           67       0      104       308,
```

Figure 10: Confusion Matrix of k-NN model for Test data



```
(Predicted   Blackboard  Google  Weather  Wikipedia
 Actual
 Blackboard         436       0      564         36
 Google               0    5217        0          0
 Weather           1466       0     1463         88
 Wikipedia          116       0      324        39,
```

Figure 11: Confusion Matrix of logistic regression model for Test data

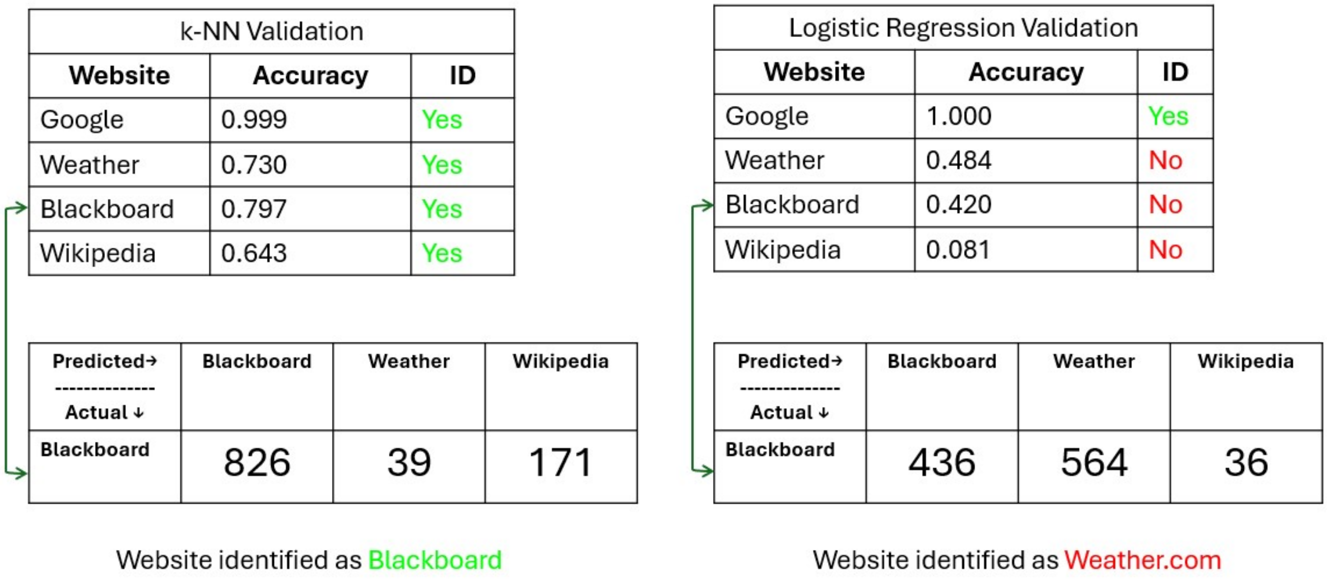
and a confusion matrix which demonstrates the majority identification. The comparison of these two models clearly shows k-NN as the superior model for identifying network traffic. The high accuracy of identifying individual packets allowed for effective classification of the source website. In figure 11, the confusion matrix for the logistic regression model shows that the model struggles with classifying Weather.com and Blackboard. The k-NN model is much better at discerning between the two websites as shown in figure 10.

5 FUTURE WORK

Despite the performance of the k-NN model, the project could be improved or adapted in many ways:

5.1 Number of Features

The model only focused on three features, which provides a strong foundation for the model, but the low number of features could limit the ability of the model to classify correctly. I would like to compare the accuracy of models using more features to my current model.



k-NN Validation		
Website	**Accuracy**	**ID**
Google	0.999	Yes
Weather	0.730	Yes
Blackboard	0.797	Yes
Wikipedia	0.643	Yes

Predicted→ ------------- Actual ↓	Blackboard	Weather	Wikipedia
Blackboard	826	39	171

Website identified as Blackboard

Logistic Regression Validation		
Website	**Accuracy**	**ID**
Google	1.000	Yes
Weather	0.484	No
Blackboard	0.420	No
Wikipedia	0.081	No

Predicted→ ------------- Actual ↓	Blackboard	Weather	Wikipedia
Blackboard	436	564	36

Website identified as Weather.com

Figure 12: Table Showing Accuracy of Each Model

5.2 Number of Websites

I only chose four websites for this project. The internet has countless websites with diverse network traffic. By including more websites, the model could be made more applicable to real-world uses. The number of websites could also affect accuracy. For example, Google uses the QUIC protocol exclusively which makes it easy to classify. If another website is included that uses the same protocol, the accuracy of identifying Google correctly would decrease.

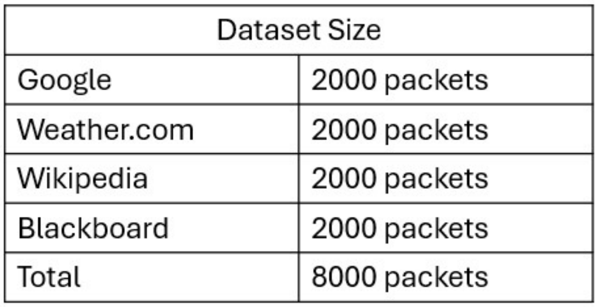
5.3 Amount of Validation Data

I only tested one validation data set for each website. By testing more thirty-second network traffic streams, the accuracy of the models could change.

5.4 Evaluation

Accuracy was the only metric used to evaluate the models. By including more standard metrics such as F1-score it could shine more light on the performance of each model.

5.5 Dataset Size



Dataset Size	
Google	2000 packets
Weather.com	2000 packets
Wikipedia	2000 packets
Blackboard	2000 packets
Total	8000 packets

Figure 13: Table Showing Size of Dataset Used to Create the Models

As shown as in figure 13, my initial dataset only included two thousand packets from each website for a total of eight thousand packets. Increasing the amount of network traffic captured might increase the accuracy of the models.

5.6 Practical Applications

Future research could focus on integrating the model into real-world applications such as systems for network monitoring, management, and security. This would demonstrate if the model is practical beyond academic contexts.

6 CONCLUSION

In this project, I developed two models for classifying network traffic by its source website. These models were trained on data from a diverse set of websites. Through the evaluation of the models, I determined that the k-NN model was superior and outperformed the logistic regression model. The performance of the k-NN model in this project is excellent since it was able to identify the source website four out of four times.

The k-NN model shows promise for future development and shows that a simple model can be used for network traffic analysis and classification which has real world applications in network security, monitoring, and management [2].

REFERENCES

[1] Anthology. 2024. *Blackboard.* https://www.blackboard.com/

[2] Christian et al Callegari. 2021. Explainable Internet Traffic Classification. In *Applied sciences 11.10 (2021): 4697-. Web.*

[3] Wikimedia Foundation. 2024. *Wikipedia.* https://www.wikipedia.org/

[4] Wireshark Foundation. 2024. *Wireshark.* https://www.wireshark.org/

[5] Allen Media Group. 2024. *The Weather Channel.* https://weather.com/

[6] IBM. 2024. *What is logistic regression?* https://www.ibm.com/topics/logistic-regression

[7] IBM. 2024. *What is the K-nearest neighbors algorithm?* https://www.ibm.com/topics/knn

[8] Google LLC. 2024. *Google.* https://www.google.com/