DARKNET TRAFFIC ANALYSIS INVESTIGATING THE IMPACT OF MODIFIED TOR TRAFFIC ON ONION SERVICE TRAFFIC CLASSIFICATION

[1]B.S.S Srikar, Dept. of CSE, bssuryasrikar2002@gmail.com

[2]Dr. A Sudhir Babu, Dept. of CSE, Professor, asudhirbabu@kluniversity.in

[1, 2]Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India.

Abstract: The review centers around analyzing network traffic in darknet conditions — like the Tor organization — to figure out what changes to traffic affect Onion Service traffic classification. It accentuates the need of further developed instruction and oversight regardless of whether Tor and Onion Administrations are intended for secrecy and security and can be utilized inappropriately. Three principal targets of the review are to find Onion Service traffic inside Tor traffic, assess the effects of traffic changes, and feature critical viewpoints in the order cycle. With a particularly eye toward network traffic design research inside the Tor organization, the venture unquestionably utilizes ML and information logical strategies to arrive at its objectives. The consequences of the undertaking could influence network observing, security, and protection since they underscore the cautious harmony between saving client security and ensuring network security.

"Index terms - *Traffic classification, machine learning, onion services, tor, anonymity, feature selection".*

1. INTRODUCTION

Tor [1] is a namelessness network that courses correspondence across a few mediators' hubs in this way disguising client personality. Tor additionally upholds the utilization of onion as the high level area name for unknown administrations once in a while alluded to as covered up administrations. Security experts, network protectors, and policing have been motivated by Pinnacle's ability to work as a restriction evasions instrument to detect Tor traffic from both other scrambled and non-encoded traffic [2], [3]. [3], [4] meant to order Tor traffic from non-Tor Traffic, [2], [5] planned to arrange the application sorts in network traffic, and [6] expected to group network traffic from other obscurity network traffic including I2P traffic and Webmix Traffic. In any case, in this work, we need to research traffic examination based noticeability of Onion Administration traffic from customary Tor traffic. Our work begins from three exploration questions we create as a premise.

Unlawful sites have been facilitated on Onion Administrations; all the more of late, they have been utilized as Command and Control (C&C) servers for botnets [7], [8]. From the angle of state run administrations and policing, then, they wish to screen and limit the Onion Administration traffic and close such administrations [9]. Limited admittance to such sites could likewise be useful in any event, for organizations to watch their frameworks from conceivable unfortunate

entertainers (like programmers) and assaults. Hence, two key reasons can make having strategies for spotting Onion Administration traffic helpful: 1. Such techniques can act as venturing stones for fingerprinting of Onion Administrations. 2. They can assist with restricting Onion Administration traffic on private and sensitive frameworks.

Tor can be utilized with specific strategies to adjust its traffic stream. Among such methodologies are presenting cushioning [10], utilizing misleading explodes and delays [11], and separating the traffic [12]. These techniques have been made fully intent on concealing Tor traffic's data spills. The essential meaning of answering RQ2 is that it will assist us with checking whether our RQ1 results will be substantial as and when such changes are executed into the Tor traffic. Would it be advisable for us we have the option to in any case recognize Onion Administration traffic, it recommends that these progressions are not effective in covering Onion Administration traffic would it be advisable for them they be acknowledged from now on. Should the progressions impact the Onion Administration classifiability, it calls into uncertainty the legitimacy of prior endeavors incorporating [3] and [6] in a climate with those alterations applied.

We subsequently focused on highlights underlining time insights. (ii) We additionally utilize components with a laid out history of performing actually in network traffic revealing patterns [13]. By and by, we do tests to survey the classifier execution with different component mixes and apply three element choice ways to deal with conclude which highlights have a more grounded relationship with the traffic sorts utilized in our work.

2. LITERATURE SURVEY

Tor is a circuit-based low-dormancy unknown specialized device that we present here. By coordinating total forward secrecy, clog control, registry waiters, respectability checking, customizable leave strategies, and a reasonable engineering for area stowed away administrations through rendezvous focuses, this second-generation Onion Routing system settles imperatives in the first plan. One [1]. Tor offers a sensible split the difference between obscurity, ease of use, and effectiveness; deals with this present reality Web; requires no specific honors or part changes; requires little synchronizing or planning between hubs. We rapidly go over our collaborations with an overall organization containing more than thirty hubs. We envelop with a stock of irritating issues by unknown correspondence.

Low-inactivity secrecy protecting organization Tor allows its clients to watch their web-based security. It involves a huge number of day to day clients served by volunteer run switches spread all over the planet. Tor experiences execution issues coming about because of clog and a low transfer to-client proportion that could hinder its overall acknowledgment and produce a general more terrible obscurity to all clients. We characterize a few sorts of administration for Pinnacle's traffic to build its exhibition. [1] Albeit most network traffic is intuitive web perusing, we recognize that a minuscule piece of mass downloads utilize an unjustifiable portion of Tor's restricted data

transfer capacity. Also, these traffic classifications have different length and data transmission limits; thus, they ought not be offered a similar “Quality of Service (QoS)”, as Tor gives now. We present and survey an machine-learning based strategy that involves application progressively to order Tor's encoded circuits and subsequently relegates different class of administration to each application. Our outcomes confirm that we can arrange made circuits on the live Tor network with extremely high accuracy more than 95% [13]. We demonstrate the way that our continuous order in mix with QoS can fundamentally upgrade the experience of Tor clients since our essential methodologies produce a 75% improvement in responsiveness and a 86% decrease in download times at the middle for intuitive clients.

Albeit various investigations have zeroed in on rush hour gridlock classificing, the quick advancement of Internet providers and the far reaching utilization of encryption present an open test [12, 14]. Security of Web clients' protection relies upon encryption, a crucial innovation applied in the few security improving gadgets that have recently surfaced. One of the most popular among them is Tor, which encodes the traffic between the shipper and the beneficiary and channels it across a dispersed organization of PCs in this way isolating them. We report atime examination on Tor traffic streams in this work [3], got between the client and the entry hub. Two situations are characterized: one to find Tor traficial streams and the other to distinguish the kind of utilization utilized: perusing, visit, streaming, mail, voip, P2P or document move. Besides, we present the Tor named dataset we made and applied to assess our classifiers in this paper.

Traffic examination and arrangement are turning out to be vital for powerful asset distribution and organization the board since the volume of organization traffic [13] is extending dramatically. Be that as it may, with creating security innovation, scrambled correspondence — one of the most frequently utilized encryption techniques — is making this work really testing. Tor [4] This article presents a convolutional brain network model [3] and hexadecimal crude bundle header based strategy for Tor traffic grouping. Our strategy shows an extremely high precision when contrasted with serious ML methods. We use UNB-CIC Tor network traffic information to approve our methodology freely. Our technique shows 99.3% accuracy for the fractionized Tor/non-Tor traffic characterization in view of the tests.

Prestigious mysterious correspondence framework Tor is utilized to safeguard clients' internet based protection. It upholds TCP applications and packs application information into encoded equivalent estimated cells to disguise some confidential data of clients, such the running application type (Web, P2P, FTP, Others). Destructive applications are notable since they can assist with bringing down the obscurity set and backing different assaults. In any case, the current Tor configuration can't conceal some application exercises [5]. P2P applications, for example, commonly transfer and download documents simultaneously, and Tor traffic similarly protects this social trademark. Propelled by this knowledge, we take a gander at another attack on Tor in light

of traffic order that can recognize application sorts from Tor traffic. Initial, an aggressor intentionally picks different stream qualities — like burst volumes and headings — to mirror the application ways of behaving and utilizes a compelling ML technique to demonstrate numerous sorts of utilizations. Then, at that point, target's Tor traffic might be arranged utilizing these known models and their application type can be found. Our investigations affirm the practicality and proficiency of the traffic order assault we have applied on Tor.

3. METHODOLOGY

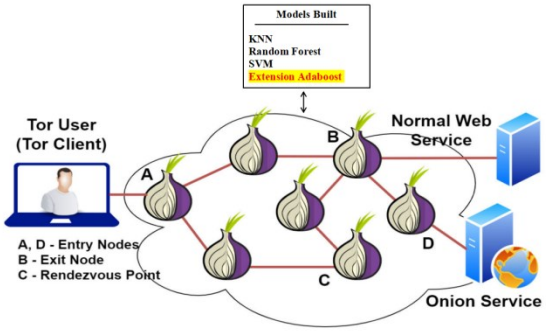
i) Proposed Work:

ADABOOST is introduced as an improvement from the conventional framework. Through extraordinary precision in arranging network traffic into Tor and Onion administrations, ADABOOST works on the presentation of the framework. It shows preferable execution over the acknowledged ML methods applied in the ordinary framework. Besides, ADABOOST extraordinarily expands the accuracy of order, subsequently unparalleled ordinary strategies in steadfastness and precision. Especially zeroing in on Onion Administrations, the task named expansions coordinate the Adaboost classifier to accomplish a huge close to 100% exactness in working on the grouping of Tor Traffic [14, [15]. Traffic order precision inside the Tor network is greatly upgraded by this element. Through an easy to use Flask system with SQLite combination, join and sign-in strategies for client testing are smoothed out, in this way working on commonsense utility. Keeping up with client protection and organization security, this ensures a perfect and safe experience, thus the structure is effectively accessible for functional Darknet Traffic Analysis. Introduced as an improvement from the customary framework is ADABOOST. Through extraordinary accuracy in grouping network traffic into Tor and Onion administrations, ADABOOST works on the presentation of the framework. It shows preferable execution over the acknowledged machine learning techniques applied in the conventional system. Besides, ADABOOST enormously expands the precision of order, thusly unbelievable traditional strategies in reliability and exactness. Especially zeroing in on Onion Administrations, the venture named expansions coordinate the Adaboost classifier to accomplish a huge close to 100% accuracy in working on the grouping of Tor Traffic [14, [15]. Traffic grouping accuracy inside the Tor network is greatly improved by this component. Through an easy to understand Carafe structure with SQLite mix, join and sign-in strategies for client testing are smoothed out, subsequently working on down to earth utility. This ensures an impeccable and safe experience, thus the system is accessible for helpful Darknet Traffic Examination even as client obscurity and organization security are kept up with.

ii) System Architecture:

The project's architecture system comprises in a Tor client beginning the method. At first marks of access into the Tor network, the Tor client's information passes by means of Entry Nodes (A and D). From that point forward, the traffic utilizes a B-Leave Hub to leave the Tor network and arrive at the normal web. For Onion administrations, safe

client correspondence relies upon a C-Meeting Point [1, 18]. The framework conveys inside the Tor network with both normal web administrations and extraordinary Onion administrations. The models utilized — KNN [3], Random Forest, SVM, and the extension of Adaboost — have a significant impact in recognizing and fathoming the impact of changed Tor traffic on Onion Service Traffic. This comprehensive plan ensures an exhaustive exploration of the elements of darknet traffic, especially with respect to the characterization of Onion administrations, thusly giving quick examination of the points of the venture.



“Fig 1 Proposed architecture”

iii) Dataset collection:

WTFPAD [10] Dataset - This dataset comprises of organization traffic changed with WTF-PAD to assess what it means for Tor and Onion Service traffic order.

	0	1	2	3	4	5	6	7	8	9	...	290	291	292	293	294	295	296
0	1.0	1.0	-1.0	1.0	-1.0	1.0	1.0	-1.0	1.0	-1.0	...	-1.0	1.0	1.0	1.0	-1.0	-1.0	-1.0
1	1.0	-1.0	1.0	-1.0	1.0	-1.0	1.0	-1.0	1.0	-1.0	...	-1.0	1.0	1.0	1.0	-1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	-1.0	1.0	1.0	-1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
3	1.0	-1.0	1.0	1.0	-1.0	-1.0	1.0	-1.0	1.0	-1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
4	1.0	-1.0	1.0	-1.0	1.0	-1.0	1.0	-1.0	1.0	-1.0	...	1.0	-1.0	1.0	1.0	-1.0	1.0	-1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9495	1.0	1.0	1.0	1.0	-1.0	-1.0	1.0	1.0	-1.0	-1.0	...	1.0	-1.0	1.0	-1.0	1.0	1.0	-1.0
9496	-1.0	-1.0	1.0	-1.0	1.0	-1.0	1.0	-1.0	1.0	-1.0	...	1.0	1.0	-1.0	1.0	-1.0	1.0	-1.0
9497	-1.0	1.0	-1.0	1.0	-1.0	1.0	-1.0	1.0	-1.0	1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
9498	-1.0	1.0	-1.0	-1.0	1.0	-1.0	1.0	-1.0	1.0	1.0	...	-1.0	1.0	-1.0	1.0	-1.0	1.0	-1.0
9499	1.0	1.0	-1.0	1.0	-1.0	-1.0	1.0	-1.0	-1.0	-1.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0

9500 rows × 300 columns

“Fig 2 WTFPAD dataset”

No Defense Dataset - This dataset gives a gauge to correlation with survey the outcome of security strategies in the venture since it shows network traffic liberated from specific security assurances.

	0	1	2	3	4	5	6	7	8	9	...	290	291	292	293	294	295	296
0	-1.0	1.0	-1.0	-1.0	1.0	1.0	1.0	-1.0	-1.0	-1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
1	1.0	-1.0	1.0	-1.0	1.0	-1.0	1.0	1.0	-1.0	-1.0	...	1.0	1.0	-1.0	1.0	1.0	-1.0	-1.0
2	1.0	-1.0	1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	...	-1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	-1.0	1.0	1.0	1.0	-1.0	1.0	1.0	-1.0	1.0	1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
4	1.0	-1.0	1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9495	1.0	-1.0	1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
9496	-1.0	-1.0	1.0	1.0	1.0	1.0	-1.0	-1.0	-1.0	-1.0	...	1.0	1.0	1.0	1.0	-1.0	-1.0	1.0
9497	-1.0	1.0	1.0	-1.0	1.0	-1.0	-1.0	1.0	1.0	-1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
9498	1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	1.0	1.0	...	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
9499	-1.0	1.0	1.0	-1.0	1.0	1.0	-1.0	1.0	-1.0	1.0	...	-1.0	-1.0	-1.0	-1.0	1.0	-1.0	1.0

9500 rows × 300 columns

“Fig 3 No Defense dataset”

iv) Data Processing:

Data processing is transforming natural information into business esteem adding information. Information researchers for the most part handle information — that is, assemble, organize, clean, approve, dissect, and make an interpretation of data into justifiable structures like diagrams or papers. Three methods — manual, mechanical, and electronic — permit one to deal with information handling. The expectation is to raise the worth of information and straightforwardness direction. This assists organizations with running better and pursue speedy key decisions. This is profoundly affected via robotized information handling advances including PC programming. Big data among different volumes of information can be transformed into important experiences for quality control and decision-making.

v) Feature selection:

The method involved with isolating the most predictable, non-repetitive, and appropriate highlights to apply in model structure is known as element choice. As the extension and assortment of datasets continue extending, deliberately contracting their size is indispensable. Highlight choice for the most part intends to bring down displaying computational expense and upgrade the presentation of a prescient model.

One of the vital components of element designing, include determination is the method involved with picking the main highlights to enter machine learning systems. Through excess or superfluous element end and component reducing the list of capabilities to those generally relevant to the machine learning model, highlight choice strategies help to bring down the quantity of information factors. Doing highlight determination somewhat early has for the most part benefits over permitting the machine learning model figure out which elements are generally applicable.

vi) Algorithms:

K-Nearest Neighbors Nearly to other straightforward arrangement algorithms, (KNN) bunches information focuses in view of vicinity. It allots "closeness" between tests to the most frequently happening class utilizing attributes from network traffic. In spite of the fact that KNN can oversee high-layered information and handle convoluted information linkages, issues incorporate picking the appropriate worth of 'k' and taking care of [13]

```
#train KNN algorithm on No-Defence Dataset
#defining KNN tuning parameters
tuning_param = {'n_neighbors' : [1], 'p' : [1]}
knn_no_defence = GridSearchCV(KNeighborsClassifier(), tuning_param, cv=
start = timeit.default_timer()
knn_no_defence.fit(no_def_X, no_def_Y)#now train KNN
end = timeit.default_timer()
predict = knn_no_defence.predict(def_X_test) #perfrom prediction on tes
end1 = timeit.default_timer()
calculateMetrics("Original (No Defence) KNN", predict, def_y_test, (enc
```

“Fig 4 KNN”

Random Forest is a technique for group learning by which a few decision trees are joined for figures. Each tree is prepared on random information subsets with substitution (packing), and votes from every one of the trees decide a definitive figure. It blends a few trees to further develop accuracy, subsequently it's fitting for sorting out network traffic. It diminishes overfitting, gives include significance experiences, and is solid dealing with high-layered information.

```
#train Random Forest algorithm on No-Defence Dataset
#defining Random Forest tuning parameters
tuning_param = {'n_estimators' : [90]}
rf_no_defence = GridSearchCV(RandomForestClassifier(), tuning_param, cv
start = timeit.default_timer()
rf_no_defence.fit(no_def_X, no_def_Y)#now train KNN
end = timeit.default_timer()
predict = rf_no_defence.predict(def_X_test) #perfrom prediction on test
end1 = timeit.default_timer()
calculateMetrics("Original (No Defence) Random Forest", predict, def_y_
```

“Fig 5 Random forest”

Support Vector Machines (SVM) [3] succeeds in circumstances with fluctuating class borders since it is a supervised learning method looking through an ideal hyperplane in high-layered space to

productively isolate different information classes. It's applicable for gathering network traffic, particularly in twofold or multi-class settings. SVM characterizes a reasonable edge, handles high-layered information, yet may track down trouble with non-straightly distinct information and calls for mindful piece capability choice.

```
#train SVM algorithm on No-Defence Dataset
#defining SVM tuning parameters
tuning_param = {'C' : [100], 'kernel': ['rbf']}
svm_no_defence = GridSearchCV(svm.SVC(), tuning_param, cv=5)#defining s
start = timeit.default_timer()
svm_no_defence.fit(no_def_X, no_def_Y)#now train KNN
end = timeit.default_timer()
predict = svm_no_defence.predict(def_X_test) #perfrom prediction on tes
end1 = timeit.default_timer()
calculateMetrics("Original (No Defence) SVM", predict, def_y_test, (enc
```

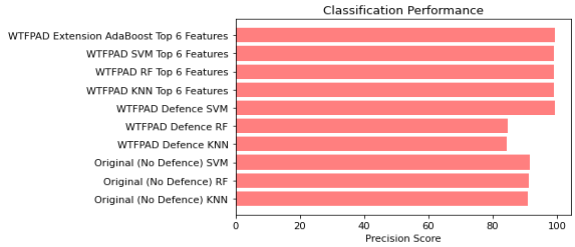
"Fig 6 SVM"

4. EXPERIMENTAL RESULTS

Precision: Precision measures among the ones sorted as positives the negligible portion of appropriately arranged occasions or tests. The recipe to decide the Precision then, at that point, is:

"Precision = True positives/ (True positives + False positives) = TP/(TP + FP)"

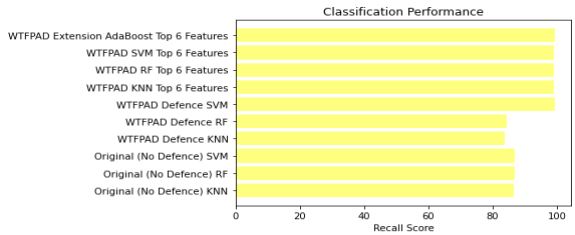
$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$



"Fig 7 Precision comparison graph"

Recall: In machine learning, recall is a measurement checking a model's ability to track down all relevant occurrences of a given class. It offers data on the culmination of a model with regards to precisely anticipated positive perceptions to the generally genuine positives.

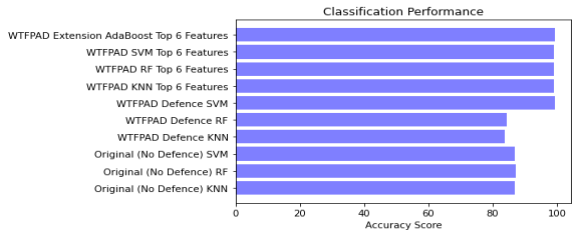
$$Recall = \frac{TP}{TP + FN}$$



"Fig 8 Recall comparison graph"

Accuracy: In a characterization work, accuracy is the level of precise expectations, subsequently measuring the overall exhibition of the forecasts of a model.

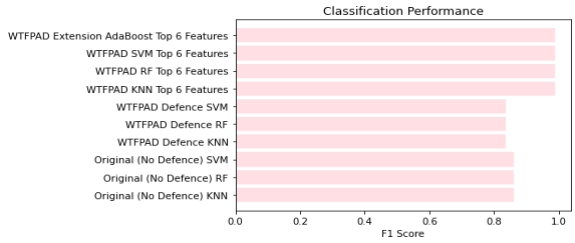
$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$



"Fig 9 Accuracy graph"

F1 Score: Reasonable for imbalanced datasets, the F1 Score is the consonant mean of precision and recall, giving a decent evaluation thinking about both false positives and false negatives.

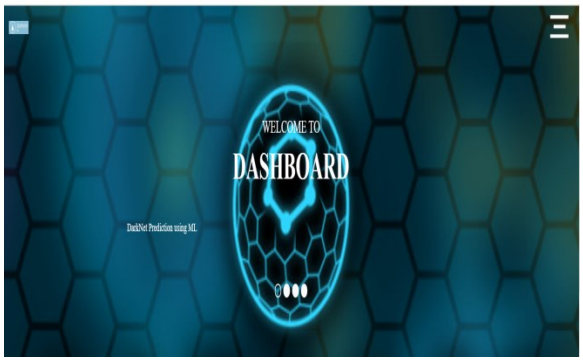
$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$



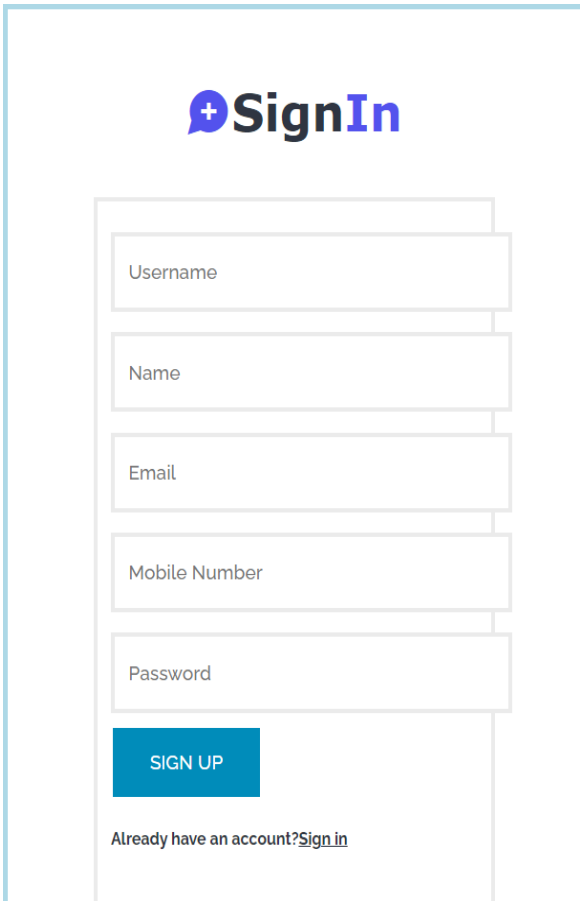
"Fig 10 F1Score"

ML Model	Accuracy	f1_score	Recall	Precision
Original (No Defence) KNN	0.868	0.860	0.868	0.919
Original (No Defence) RF	0.868	0.860	0.868	0.919
Original (No Defence) SVM	0.868	0.860	0.868	0.919
WTFPAD Defence KNN	0.838	0.838	0.838	0.845
WTFPAD Defence RF	0.838	0.838	0.838	0.845
WTFPAD Defence SVM	0.838	0.838	0.838	0.845
WTFPAD KNN Top 6 Features	0.990	0.990	0.990	0.990
WTFPAD RF Top 6 Features	0.990	0.990	0.990	0.990
WTFPAD SVM Top 6 Features	0.990	0.990	0.990	0.990
Extension WTFPAD Extension AdaBoost Top 6 Features	0.990	0.990	0.990	0.990

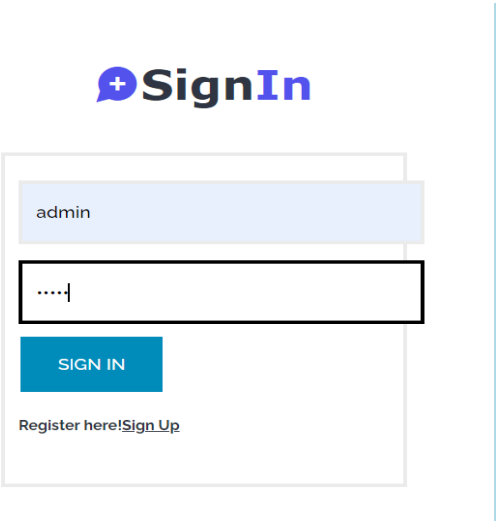
"Fig 11 Performance Evaluation "



"Fig 12 Home page"



"Fig 13 Signin page"



"Fig 14 Login page"

F1

F2

F3

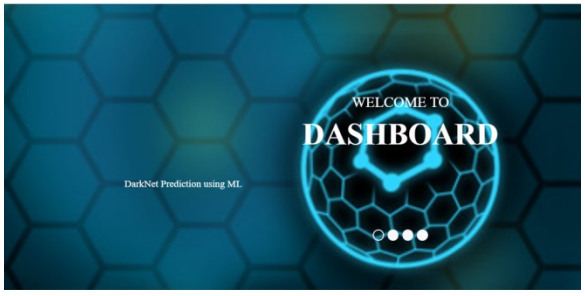
F4

F5

F6

Predict

"Fig 15 User input"



Result: **OS!**

"Fig 16 Predict result for given input"

5. CONCLUSION

The undertaking actually investigates what changed Tor traffic means for Onion Service traffic arrangement. Through intensive examination, it uncovers that changes to Tor traffic obviously impact the order accuracy. Understanding the solidness of Onion Administration traffic order under various circumstances relies upon this acknowledgment, which additionally assists with growing information on darknet traffic elements ([3], [6]). For the order task, the venture finds and positions the main element mixes. Through the distinguishing proof of significant blends, the examination explains the elements affecting Onion Administration traffic characterization accuracy above all. This data assists with further developing models and give a superior consciousness of the complicated connections inside the dataset. Broad examination is completed to get a handle on what different perspectives mean for classifiers and perform. This comprehensive review offers savvy data on how various attributes support the noticed presentation varieties among a few classifiers. Dependability and interpretability of the order models improve when one knows about the elements of component execution. Utilizing gathering strategies — all the more particularly, Adaboost — the undertaking grows its ability to arrive at high accuracy in Onion Service traffic arrangement. Adaboost expands the overall accuracy and versatility of the characterization framework by amassing the gauges of a few separate models. This drive upholds the target of the task — that of giving a refined and precise answer for darknet traffic examination. The task joins a pleasant Carafe interact with safe validation

to upgrade the entire client experience during framework testing. This connection point ensures a protected environmental elements for information entering for execution assessment and improves on client cooperations. The undertaking is effectively available and down to earth for testing and true purposes since the blend of safety and ease of use observes best guidelines in framework design.

6. FUTURE SCOPE

Further concentrate on classifier execution when Tor traffic is intentionally different with instruments like Traffic Sliver or WTFPAD [10] is urgent to assess the impacts of Tor traffic adjustments. Such changes can impact the ability to separate Tor from non-Tor traffic, thusly influencing network observing and security. The review can investigate how different attributes impact classifier execution for the ID of Onion Service traffic. This examination gives comprehension of what the attributes mean for the classification accuracy. More review is totally important to comprehend the ramifications of rapidly spotting Tor and Onion Service activity ([3], [6]). Given the bigger social and security outcomes, this information ought to cover traffic observation as well as could be expected government and delicate organization limitations. Indeed, even within the sight of muddling techniques, the examination could research state of the art innovations to work on the accuracy of spotting Onion Administration traffic. This work could move imaginative ways to deal with support Tor network order accuracy.

REFERENCES

[1] R. Dingledine, N. Mathewson, and P. Syverson, ‘‘Tor: The secondgeneration onion router,’’ in Proc. 13th USENIX Secur. Symp. (SSYM), San Diego, CA, USA, Aug. 2004, pp. 303–320.

[2] M. Al Sabah, K. Bauer, and I. Goldberg, ‘‘Enhancing Tor’s performance using real-time traffic classification,’’ in Proc. ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, Oct. 2012, pp. 73–84.

[3] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, ‘‘Characterization of Tor traffic using time based features,’’ in Proc. 3rd Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), Porto, Portugal, Feb. 2017, pp. 253–262.

[4] M. Kim and A. Anpalagan, ‘‘Tor traffic classification from raw packet header using convolutional neural network,’’ in Proc. 1st IEEE Int. Conf. Knowl. Innov. Invention (ICKII), Jeju Island, South Korea, Jul. 2018, pp. 187–190.

[5] G. He, M. Yang, J. Luo, and X. Gu, ‘‘Inferring application type information from Tor encrypted traffic,’’ in Proc. 2nd Int. Conf. Adv. Cloud Big Data (CBD), Washington, DC, USA, Nov. 2014, pp. 220–227.

[6] A. Montieri, D. Ciuonzo, G. Aceto, and A. Pescapé, ‘‘Anonymity services tor, I2P, JonDonym: Classifying in the dark (web),’’ IEEE Trans. Dependable Secure Comput., vol. 17, no. 3, pp. 662–675, May 2020.

[7] (May 2017). WCry Ransomware Analysis. Accessed: Apr. 26, 2023. [Online]. Available: https://www.secureworks.com/research/wcryransomware-analysis

[8] (Jul. 2019). Keeping a Hidden Identity: Mirai C&Cs in Tor Network. Accessed: Apr. 26, 2023. [Online]. Available: https://blog.trendmicro.com/trendlabs-security-intelligence/keeping-a-hidden-identity-mirai-ccsin-tor-network/

[9] (Nov. 2014). Global Action Against Dark Markets on Tor Network. Accessed: Aug. 4, 2020. [Online]. Available: https://www.europol.europa.eu/newsroom/news/global-action-against-dark-markets-tornetwork

[10] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, ''Toward an efficient website fingerprinting defense,'' in Proc. 21st Eur. Symp. Res. Comput. Secur. (ESORICS), Heraklion, Greece, Sep. 2016, pp. 27–46.

[11] T. Wang and I. Goldberg, ''Walkie-talkie: An efficient defense against passive website fingerprinting attacks,'' in Proc. 26th USENIX Secur. Symp. (SEC), Vancouver, BC, Canada, Aug. 2017, pp. 1375–1390.

[12] W. De la Cadena, A. Mitseva, J. Hiller, J. Pennekamp, S. Reuter, J. Filter, T. Engel, K. Wehrle, and A. Panchenko, ''TrafficSliver: Fighting website fingerprinting attacks with traffic splitting,'' in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, Nov. 2020, pp. 1971–1985.

[13] J. Hayes and G. Danezis, ''k-fingerprinting: A robust scalable website fingerprinting technique,'' in Proc. 25th USENIX Conf. Secur. Symp. (SEC), Austin, TX, USA, Aug. 2016, pp. 1187–1203.

[14] X. Bai, Y. Zhang, and X. Niu, ''Traffic identification of Tor and webmix,'' in Proc. 8th Int. Conf. Intell. Syst. Design Appl. (ISDA), Kaohsiung, Taiwan, vol. 1, Nov. 2008, pp. 548–551.

[15] O. Berthold, H. Federrath, and S. Köpsell, ''Web MIXes: A system for anonymous and unobservable Internet access,'' in Proc. Int. Workshop Design Issues Anonymity Unobservability, in Lecture Notes in Computer Science, vol. 2009, H. Federrath, Ed., Berkeley, CA, USA, Jul. 2000, pp. 115–129.

[16] B. Zantout and R. Haraty, ''I2P data communication system,'' in Proc. 10th Int. Conf. Netw. (ICN), Sint Maarten, The Netherlands, Jan. 2011, pp. 401–409.

[17] P. Sirinam, M. Imani, M. Juarez, and M. Wright, ''Deep fingerprinting: Undermining website fingerprinting defenses with deep learning,'' in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), Toronto, ON, Canada, Oct. 2018, pp. 1928–1943.

[18] R. Overdorf, M. Juárez, G. Acar, R. Greenstadt, and C. Díaz, ''How unique is your.onion?: An analysis of the fingerprintability of Tor onion services,'' in Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS), Dallas, TX, USA, Oct. 2017, pp. 2021–2036.

[19] I. H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann, 2011.

[20] X. He, D. Cai, and P. Niyogi, ''Laplacian score for feature selection,'' in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Vancouver, BC, Canada, Dec. 2005, pp. 507–514.

[21] M. Gan and L. Zhang, ‘‘Iteratively local Fisher score for feature selection,’’ Appl. Intell., vol. 51, pp. 6167–6181, Aug. 2021.