**DARKWEB CLASSIFIER USING MACHINE LEARNING TECHNIQUE**

## **A PROJECT REPORT**

Submitted in partial fulfillment of the requirement for the award of the degree

of

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE AND ENGINEERING**

**SUBMITTED BY**

MAYANK(21103088)

HARSHIT JAIN(21103060)

Under the supervision of

**Dr. Urvashi**

**Assistant Professor**

****

**Department of Computer Science and Engineering**

**Dr. B. R. Ambedkar National Institute of Technology Jalandhar**

**-144008, Punjab (India)**

**May 2024**

**Dr. B. R. Ambedkar National Institute of Technology Jalandhar**

**CANDIDATES’ DECLARATION**

We hereby certify that the work presented in this project report entitled “**DARKWEB CLASSIFIER USING MACHINE LEARNING TECHNIQUE**” in partial fulfillment of the requirement for the award of a Bachelor of Technology degree in Computer Science and Engineering, submitted to the Dr. B R Ambedkar National Institute of Technology, Jalandhar is an authentic record of our own work carried out during the period from July 2023 to May 2024 under the supervision of Dr. Urvashi, Assistant Professor, Department of Computer Science & Engineering, Dr. B R Ambedkar National Institute of Technology, Jalandhar.

We have not submitted the matter presented in this report to any other university or institute for the award of any degree or any other purpose.

Date: 31th May, 2024

Submitted by

MAYANK(21103088)

HARSHIT JAIN(21103060)

This is to certify that the statements submitted by the above candidates are accurate and correct to the best of our knowledge and are further recommended for external evaluation.

Dr. Urvashi, Supervisor Dr. Rajneesh Rani

Assistant Professor Head and Associate Professor

Deptt. of CSE Deptt. of CSE

**ACKNOWLEDGEMENT**

The completion of this project on "Traffic Classification of User Behaviors in Tor, I2P, ZeroNet, and Freenet" was made possible through the support and guidance of many individuals. The journey was a collaborative effort, and we are deeply grateful to everyone who contributed to its success.

We would like to express our deepest gratitude to our project mentor, Dr. Urvashi, whose belief in our vision and invaluable suggestions were instrumental in guiding us through the challenges we faced. His unwavering support and expertise were crucial in the successful completion of this project.

We extend our heartfelt thanks to Dr. Rajneesh Rani, Head of the Department of Computer Science and Engineering, for her direct and indirect support throughout this endeavor. Her encouragement and assistance were pivotal in navigating the complexities of our research.

We are also profoundly grateful to Dr Aruna Malik , Coordinator of the Major Project, for providing us with the necessary resources and continuous support. Her dedication ensured we had everything needed to accomplish our goals.

Our sincere appreciation goes to all the faculty members of the Department of [Your Department]. Their constant encouragement and guidance were vital in refining our work and pushing us to achieve excellence. We also extend our thanks to the laboratory staff for their timely support and assistance.

Lastly, we are grateful to our families and friends for their unwavering support and patience throughout this journey. Their encouragement and understanding were invaluable.

Thank You.

[Mayank,Harshit]

**ABSTRACT**

Darkweb Analyser is a cutting-edge application designed to classify and monitor user behaviors in various anonymous networks such as Tor, I2P, ZeroNet, and Freenet. This project addresses the growing concern over illegal activities in these darknets by deploying a hierarchical classifier capable of distinguishing between different types of darknet traffic and user behaviors with remarkable accuracy. Utilizing advanced machine learning techniques, Darkweb Analyser leverages a robust dataset collected through a custom-built darknet data probe. This dataset includes traffic from multiple user activities, meticulously labeled and preprocessed to enhance the accuracy of the classification models.

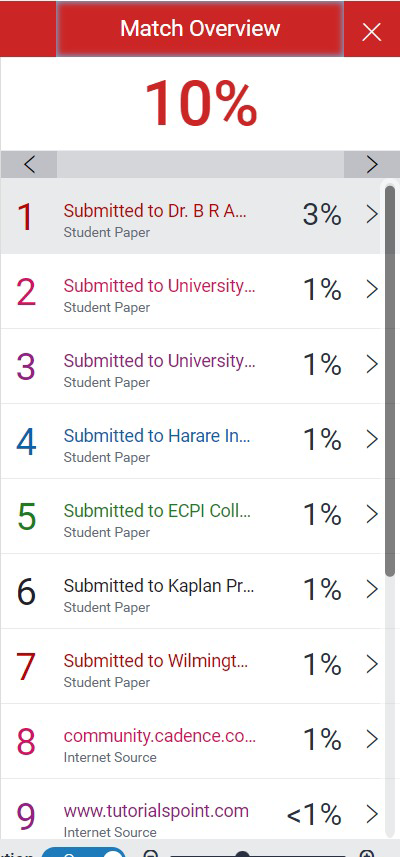
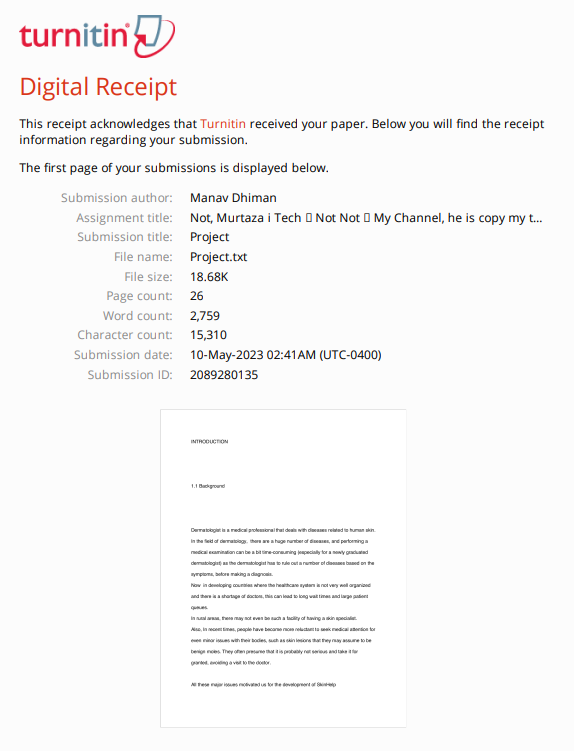
The preprocessing phase involves extracting 26 distinct time-based flow features from the traffic data, ensuring the model trains on high-quality and representative inputs. The classification framework employs a hierarchical model, comprising six local classifiers, each optimized using a variety of machine learning algorithms such as Random Forest, Gradient Boosting Decision Trees, and LightGBM. This approach not only allows for the identification of darknet traffic but also provides granular insights into user behaviors within each network.

Model training is conducted on a comprehensive dataset featuring eight types of user behaviors across the four darknets, including browsing, chat, email, and file transfer activities. The classifiers are fine-tuned to achieve high performance metrics, with the best models reaching accuracy rates of up to 99.4% for darknet type classification and 92.9% for user behavior identification. These models are deployed on a scalable backend infrastructure, enabling real-time traffic analysis and behavior classification.

Darkweb Analyser is implemented using Python and integrates seamlessly with network monitoring tools to provide a powerful solution for network administrators and law enforcement agencies. The deployment on cloud platforms ensures robust performance and scalability, allowing the system to handle extensive traffic loads efficiently. By advancing the capabilities of darknet traffic analysis, Darkweb Analyser represents a significant step forward in cybersecurity, providing essential tools to combat illegal activities and enhance the security of digital communications.

**PLAGIARISM REPORT**

We have checked plagiarism for our Project Report for our project a **Turnitin.** We are thankful to our mentor Dr. Samayveer Singh for guiding us at this. Below is the digital receipt. The Plagiarism is approximately 10%.

****

**TABLE OF CONTENTS**

|  | CANDIDATES’ DECLARATION  ACKNOWLEDGEMENT  ABSTRACT  PLAGIARISM REPORT | I  II  III  IV |
| --- | --- | --- |
| 1. | INTRODUCTION   * 1. Background of the Problem   2. Literature Survey   3. Problem Statement   4. Motivation   5. Feasibility   6. Research Objectives | 6  6  7  7  8  9 |
| 2. | PROPOSED SOLUTION | 10-13 |
| 3. | TECHNOLOGY ANALYSIS   * 1. Tech Stack Analysis | 14-15 |
| 4. | ECONOMIC ANALYSIS | 16 |
| 5. | RESULT AND DISCUSSION | 17-23 |
| 6. | CONCLUSION | 24 |
| 7. | REFERENCES | 25 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Background**

The rise of anonymous networks, such as Tor, I2P, ZeroNet, and Freenet, has introduced new complexities into the realm of cybersecurity. These networks provide users with anonymity and privacy, which is beneficial for protecting personal information and enabling free expression. However, these same features make these networks attractive for illicit activities, ranging from illegal drug trade and human trafficking to cyber-attacks and data breaches. This dual-use nature of anonymous networks necessitates robust mechanisms for monitoring and classifying traffic to differentiate between legitimate and illegitimate activities.

Tor (The Onion Router) is the most widely known anonymous network, employing a series of volunteer-operated servers worldwide to encrypt and route user traffic, thereby masking user identities. I2P (Invisible Internet Project) employs garlic routing, bundling multiple messages together to enhance privacy and prevent tracing. ZeroNet integrates Bitcoin's cryptographic principles with BitTorrent technology, creating a decentralized, resilient network. Freenet, one of the earliest darknets, focuses on providing censorship-resistant communication through a distributed data storage system. Each network's architecture and user base contribute to distinct traffic patterns, complicating the task of effective traffic classification.

The anonymity provided by these networks is crucial for users in oppressive regimes, journalists, and privacy-conscious individuals. However, this same anonymity facilitates illegal activities, making it challenging for authorities to monitor and prevent misuse. Effective classification of traffic in these networks is essential for maintaining security while preserving user privacy. This study proposes a hierarchical classification method to address these challenges, aiming to accurately identify user behaviors across multiple darknets.

**1.2. Literature Survey**

Extensive research has been conducted on the architecture and security of anonymous networks. Syverson et al. (2004) discussed the robustness of Tor’s onion routing and its vulnerabilities, which can be exploited for traffic analysis【3】. Fachkha and Debbabi (2015) provided a comprehensive taxonomy of darknet activities, highlighting the need for advanced classification techniques【2】. Recent studies, such as those by Montieri et al. (2019), utilized machine learning for traffic classification, achieving moderate success but lacking detailed behavior analysis and comprehensive datasets. This study builds on these efforts by employing a hierarchical classification approach and extending the scope to include multiple darknets and specific user behaviors.

Cuzzocrea et al. (2017) analyzed Tor traffic using machine learning techniques, demonstrating the potential of these methods but also revealing challenges like the need for large labeled datasets【7】. Wang et al. (2018) focused on measuring and analyzing darknet person attributes, underscoring the importance of detailed traffic analysis【1】. By combining insights from these studies, this research aims to develop a more granular and accurate classification method.

Building on previous work, this study introduces a hierarchical classifier that distinguishes traffic from four major darknets (Tor, I2P, ZeroNet, and Freenet) and identifies 25 specific user behaviors within these networks. The hierarchical classification method proposed in this research aims to improve accuracy by breaking down the classification process into three layers: identifying darknet traffic, determining the specific darknet, and classifying user behavior. This method leverages 26 time-based flow features extracted from darknet traffic to enhance the precision of traffic classification.

This approach addresses the limitations of previous studies, such as the lack of detailed behavior analysis and the reliance on limited datasets. By deploying traffic collecting probes in real darknet environments, this study captures and labels authentic darknet traffic, providing a robust dataset for training and validating the hierarchical classifier. This comprehensive dataset and innovative classification method contribute significantly to the field of darknet traffic analysis and user behavior identification.

**1.3. Problem Statement and its Necessity**

The major issues that spurred the arrangement are as following:

**Unavailability of Effective Monitoring Tools:**

Current tools for monitoring darknet traffic often lack the granularity needed to distinguish between different types of user behaviors across multiple darknets. This limitation hinders the ability to effectively identify and mitigate illicit activities.

**Complexity of Traffic Patterns:**

The unique architectures and user bases of Tor, I2P, ZeroNet, and Freenet result in diverse traffic patterns. Existing classification methods struggle to cope with this complexity, leading to inaccurate or incomplete traffic analysis.

**Need for Advanced Classification Techniques:**

Traditional traffic analysis methods are insufficient for the nuanced task of identifying specific user behaviors within darknets. Advanced machine learning techniques, particularly hierarchical classifiers, offer a promising solution by breaking down the classification task into more manageable sub-tasks.

**Enhancing Network Security**:

Accurate classification of darknet traffic can significantly enhance network security by enabling targeted interventions. This is crucial for preventing illegal activities while preserving the privacy and anonymity of legitimate users.

**1.4. Motivation**

**Improving Security and Monitoring:** The primary motivation for this research is to develop a robust method for accurately classifying darknet traffic, thereby improving the ability to monitor and secure these networks. This is essential for identifying and mitigating illicit activities while respecting user privacy.

**Advancing Machine Learning Applications:** This study aims to advance the application of machine learning techniques in traffic classification, demonstrating the effectiveness of hierarchical classifiers for complex tasks involving multiple darknets and diverse user behaviors.

**Providing Valuable Insights:** The research will provide valuable insights into the behavior of users in anonymous networks, contributing to the broader field of cybersecurity. The findings will inform the development of more effective monitoring and intervention strategies.

**1.5. Feasibility : Non-Technical and Technical**

**Technical Feasibility:**

The project leverages powerful machine learning algorithms and readily available computational resources. High-level programming languages like Python, along with machine learning libraries such as Scikit-learn, XGBoost, and TensorFlow, facilitate the development of sophisticated classification models. Cloud-based platforms like Google Colaboratory and AWS EC2 instances provide the necessary computational power, enabling the training and evaluation of complex models even on low-end personal computers.

**Social Feasibility:**

There is a growing need for effective tools to monitor and secure anonymous networks, driven by the increasing prevalence of illicit activities in these environments. The research addresses a significant gap in existing monitoring capabilities, providing a solution that enhances security without compromising user privacy.

**Economic Feasibility:**

The project is economically feasible, as it relies on open-source libraries and publicly available datasets. The primary costs are associated with computational resources, which are minimized by using cloud-based platforms. The potential benefits, including improved network security and reduced illegal activities, far outweigh the modest development expenses.

**Scope:**

This research aims to develop a hierarchical classification method capable of accurately identifying user behaviors across multiple darknets. The findings will enhance the ability to monitor and secure these networks, benefiting cybersecurity professionals and researchers.

**1.6 Research Objectives**

**Develop a Hierarchical Classifier:** To create a multi-layer classification model that can accurately distinguish between different types of darknet traffic and identify specific user behaviors within these networks.

**Leverage Advanced Machine Learning Techniques:** Employ state-of-the-art machine learning algorithms, including XGBoost and deep learning models, to achieve high accuracy and robustness in traffic classification.

**CHAPTER 2**

**PROPOSED SOLUTION**

This study addresses the classification of user behaviors in anonymous networks such as Tor, I2P, ZeroNet, and Freenet by developing a hierarchical classification model. The goal is to accurately identify and differentiate between various types of activities within these networks, thereby enhancing the ability to monitor and secure these environments.

The proposed solution in the document is a hierarchical classification method to identify user behaviors within darknet traffic. Here is a detailed summary of the proposed solution:

**Hierarchical Classification Method**

**Structure**

The hierarchical classifier is designed to manage the complexity of identifying user behaviors in different darknets (Tor, I2P, ZeroNet, Freenet) by breaking down the classification process into multiple levels:

**First Level**: Distinguishes between normal internet traffic and darknet traffic.

**Second Level:** Identifies the specific type of darknet (e.g., Tor, I2P).

**Third Level:** Classifies the user behavior within the identified darknet.

The structure allows for separate training of local classifiers at each level, which makes the training process more flexible and efficient.

**Hierarchal classifier structure:**

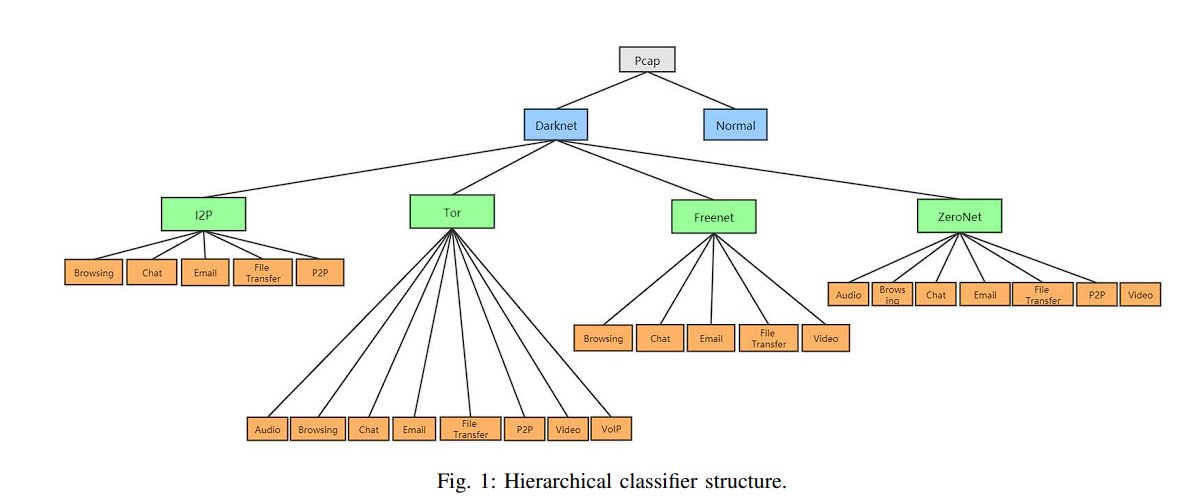


Fig.2.1 hierarchal classifier structure

**Training Methods**

Three training methods for the hierarchical classifier are considered:

Binary Classifier for Each Node: Similar to the one-versus-rest method, which has a large performance overhead.

Multi-classifier for Each Layer: This approach requires only three classifiers for the entire hierarchy but suffers from lower accuracy and applicability issues due to dissimilar behaviors among different darknets.

Multi-classifier for Each Parent Node: Each classifier is trained independently, forming a tree structure. This method is intuitive and balances performance and overhead.

For the purposes of this study, the third method was chosen due to its balanced performance and overhead considerations.

**Classification Algorithms**

The document reviews six machine learning (ML) algorithms and two deep learning (DL) algorithms for the selection of local classifiers:

**XGBoost**

**LightGBM**

Each local classifier is trained independently, and the best-performing algorithm is selected for each parent node in the hierarchical structure.

**Advantages and Disadvantages of hierarchical structure**

**Advantages:**

Flexibility in training: Each local classifier can be trained independently with the most suitable algorithm.

Improved accuracy: By breaking down the classification process, the hierarchical method can achieve higher accuracy compared to a flat multi-class classifier.

**Disadvantages:**

Error propagation: Errors in one layer can affect subsequent layers, impacting the final classification result.

**Conclusion**

The hierarchical classification method proposed in the document aims to efficiently and accurately identify user behaviors in darknet traffic by leveraging a flexible and structured training approach. Each layer in the hierarchy addresses a specific aspect of the classification, allowing for tailored algorithms and improving overall performance while managing complexity.

This approach balances the need for accuracy and efficiency, making it suitable for the challenging task of darknet traffic classification​​.

The development of the classification model involves several steps: data collection, preprocessing, model training, and deployment. The first step involves gathering a comprehensive dataset of network traffic from the four target networks. This dataset includes a diverse range of traffic patterns to ensure the model can accurately classify different user behaviors.

Once the data is collected, it is preprocessed to remove any irrelevant or redundant information and to standardize the format for analysis. This step is crucial for ensuring the quality and reliability of the training data.

Next, a hierarchical classification algorithm is trained on the preprocessed dataset. This involves using a neural network architecture to learn the features and patterns associated with different types of traffic. The model is iteratively tested and refined to improve its accuracy and robustness.

After the training process is complete, the model is deployed within a web application where users can upload network traffic data for analysis. The application utilizes the trained model to classify the traffic, providing a detailed report on the identified user behaviors. This helps security professionals monitor these networks more effectively and respond to potential threats promptly.

Overall, this project provides a comprehensive solution for traffic classification in anonymous networks, enhancing both security and user privacy. The following sections provide a detailed overview of the development process, step by step.

**User Interface:** The dashboard of the application allows users to upload network traffic data for analysis.

**Result Analysis:** Once the data is uploaded, the application analyzes the traffic and provides a detailed report on the identified behaviors.

As specified earlier, transfer learning was utilized to train the model, and a pre-trained DenseNet169 architecture was custom-trained for this purpose. The following steps outline the procedure followed to develop the model:

**1. Data Gathering**

The model was trained on a dataset that includes various types of traffic from Tor, I2P, ZeroNet, and Freenet. This dataset is collected from multiple sources to ensure diversity and comprehensiveness.

**2. Data Labeling**

The dataset is labeled to map different types of traffic to their respective behaviors. This step is essential for training the model to recognize and classify different patterns accurately.

**3. Dataset Analysis**

An analysis of the dataset reveals the distribution of different types of traffic. This analysis helps identify any imbalances in the dataset that need to be addressed to prevent overfitting.

**4. Undersampling**

To address the issue of class imbalance, random undersampling is performed on classes with an overrepresentation of samples.

**5. Loading the Data**

The processed dataset is loaded into an ImageDataBunch object, and typical data augmentations are applied. The dataset is normalized using ImageNet statistics to standardize the input for the model.

**6.Model Training and Evaluation**  
A learner object is created using the pre-trained DenseNet169 architecture. The model's training process begins with determining the optimal learning rate using a learning rate finder.

The model is then trained using the optimal learning rate, and the best model based on accuracy is saved for deployment.

The optimal model achieved an accuracy of approximately 91.2% at epoch 19, with an f-measure of 91.7%.

**7.Model Interpretation**

The confusion matrix is plotted to interpret the model's performance, providing insights into its accuracy across different classes.

**CHAPTER 3**

**TECHNOLOGY ANALYSIS**

**3.1. Tech Stack Analysis**

The technologies used in the project are:-

**3.1.1. Google Colab**

Google Colab is a web-based coding framework that enables users to write and run Python code without any setup or configuration. It offers free access to GPUs and provides easy sharing options. With Colab, users can write and execute code, build machine learning models, and collaborate with other developers on projects. It is based on Jupyter Notebooks and allows for the use of markdown cells to create formatted text alongside code. Additionally, it is entirely self-contained, meaning users do not need to install any software or manage any infrastructure to use it.

**3.1.2. Python**

Python is a widely used high-level programming language that is interpreted and interactive. It supports object-oriented programming and is well-suited for a variety of applications, including machine learning. Python has a strong library ecosystem, with extensive support for machine learning algorithms through popular libraries such as sci-kit-learn, Keras, and TensorFlow. These libraries simplify the development and management of machine learning algorithms, making it easier for developers to implement complex models and analyze large datasets.

Python is a high-level, interpreted programming language known for its simplicity and readability. Developed by Guido van Rossum and first released in 1991, Python emphasizes code readability and allows developers to use fewer lines of code to accomplish tasks compared to other programming languages like C++ or Java. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

**Key Features of Python:**

**Readability and Simplicity:** Python's syntax is designed to be readable and straightforward, making it an excellent choice for beginners and experienced developers alike. The language's simplicity allows developers to focus on solving problems rather than understanding complex syntax.

**Extensive Standard Library:** Python comes with a rich standard library that includes modules and packages for various tasks such as web development, data analysis, machine learning, and more. This extensive library reduces the need for writing code from scratch.

**Interpreted Language:** Python is an interpreted language, meaning that the code is executed line by line, which makes debugging easier. This feature also allows for rapid prototyping and iterative development.

**Dynamic Typing:** Python uses dynamic typing, which means that variable types are determined at runtime. This flexibility can speed up development but requires careful handling to avoid runtime errors.

**Versatility:** Python can be used for a wide range of applications, from web development and scientific computing to artificial intelligence and automation. Its versatility makes it a popular choice among developers across different fields.

**Community and Ecosystem:** Python has a large and active community, which contributes to a vast ecosystem of third-party libraries and frameworks. Popular libraries include NumPy for numerical computing, pandas for data manipulation, TensorFlow and PyTorch for machine learning, and Flask and Django for web development.

**Cross-Platform:** Python is a cross-platform language, meaning it can run on various operating systems such as Windows, macOS, and Linux without requiring modifications to the code.

**3.1.2 Jupyter Notebook**

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It is widely used in data science, scientific computing, and machine learning for its interactive and user-friendly interface. Jupyter Notebook supports over 40 programming languages, including Python, R, and Julia.

**Key Features of Jupyter Notebook:**

Interactive Coding: Jupyter Notebook provides an interactive environment where you can write and execute code in real-time. This feature is particularly useful for data analysis and exploration, allowing you to see the results of your code immediately.

**Rich Text Support:** In addition to code, Jupyter Notebook supports markdown for creating formatted text, mathematical equations using LaTeX, and multimedia content such as images and videos. This makes it easy to create comprehensive and well-documented reports.

**Data Visualization:** Jupyter Notebook integrates seamlessly with data visualization libraries such as Matplotlib, Seaborn, and Plotly. You can generate and display plots and charts within the notebook, making it easier to analyze and interpret data.

**Reproducible Research:** Jupyter Notebooks can be shared with others, allowing for reproducibility of research. Others can view, edit, and run the notebooks to verify and build upon your work. This feature is essential for collaboration and sharing insights within the scientific community.

**Extensibility:** Jupyter Notebook supports extensions that add additional functionality, such as interactive widgets, code formatting, and version control. These extensions enhance the user experience and provide additional tools for development and analysis.

**Integration with Other Tools:** Jupyter Notebook can be integrated with various data science and machine learning tools, such as pandas for data manipulation, scikit-learn for machine learning, and TensorFlow for deep learning. This integration makes it a powerful tool for end-to-end data science workflows.

**Cloud and Collaboration:** Jupyter Notebooks can be run on local machines or in the cloud using services like Google Colab, Microsoft Azure Notebooks, and AWS SageMaker. Cloud-based notebooks enable easy sharing and collaboration among multiple users.

**CHAPTER 4**

**ECONOMIC ANALYSIS**

Our project leverages free and secure technologies to build the application and platforms, including APIs, datasets, and dependencies. By utilizing these free software resources, we ensure that there are no costs associated with using our application, making it entirely accessible to users without financial barriers. Here’s a detailed look at the economic considerations:

**Cost-Free Technologies:** All the tools and technologies we have used in the development of our application are open-source and free to use. This includes the programming languages, libraries, and frameworks, ensuring no upfront or ongoing costs for users.

**Affordable and User-Friendly Solutions:** Our primary objective is to provide solutions that are not only affordable but also user-friendly. By using readily available tools, we ensure that our application is equipped with necessary features to address common challenges effectively without any financial burden on the users.

**Python:** Python is a widely used, high-level programming language that is open-source and free. Its strong library ecosystem, which includes libraries like TensorFlow and Keras for machine learning, allows us to develop complex models and applications without incurring any licensing fees. Python's versatility and ease of use make it an ideal choice for our project, ensuring efficient development and maintenance.

**Jupyter Notebook:** Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It is free to use and provides an interactive computing environment, which is particularly useful for data analysis and machine learning tasks. Jupyter Notebook's integration with Python makes it a valuable tool for developing and testing our models in an efficient and cost-effective manner.

**Google Colab:** Google Colab is a powerful, free tool for running Python code in the cloud. It provides free access to computational resources, including GPUs, which are essential for training machine learning models. Using Google Colab allows us to avoid the costs associated with purchasing and maintaining physical hardware.

By utilizing these cost-free and efficient tools, we ensure that the economic burden on developing and maintaining the application is minimized, allowing us to provide a robust and accessible solution to our users without any associated costs

**CHAPTER 5**

**RESULT AND DISCUSSION**

The paper details the process for generating a dataset that includes traffic from various user behaviors across four darknets: Tor, I2P, ZeroNet, and Freenet. The user behaviors considered are browsing, chat, email, audio streaming, video streaming, file transfer, P2P, and VoIP.

**Hierarchal classifier structure:**

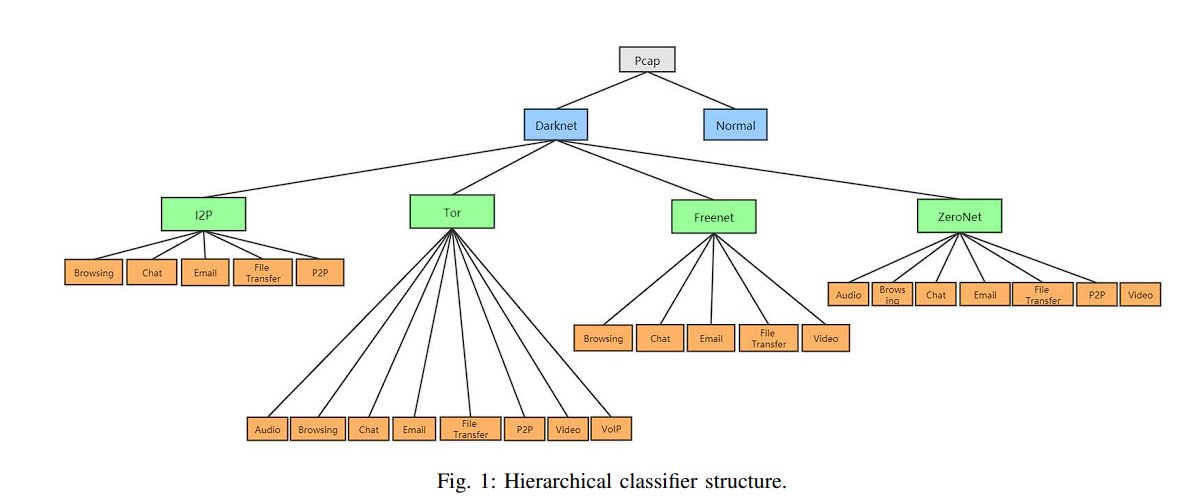


Fig.5.1Hierarchal classifier structure

**Classifier of Darknet User Behaviors**

**Hierarchical Classification Method**

The hierarchical classifier structure is designed to improve the accuracy and efficiency of identifying darknet user behaviors by breaking down the classification process into multiple layers. This approach addresses the challenge of directly training a multi-class classifier, which can result in lower accuracy due to the large number of categories.

**The hierarchical classifier consists of three layers(Fig.5.1):**

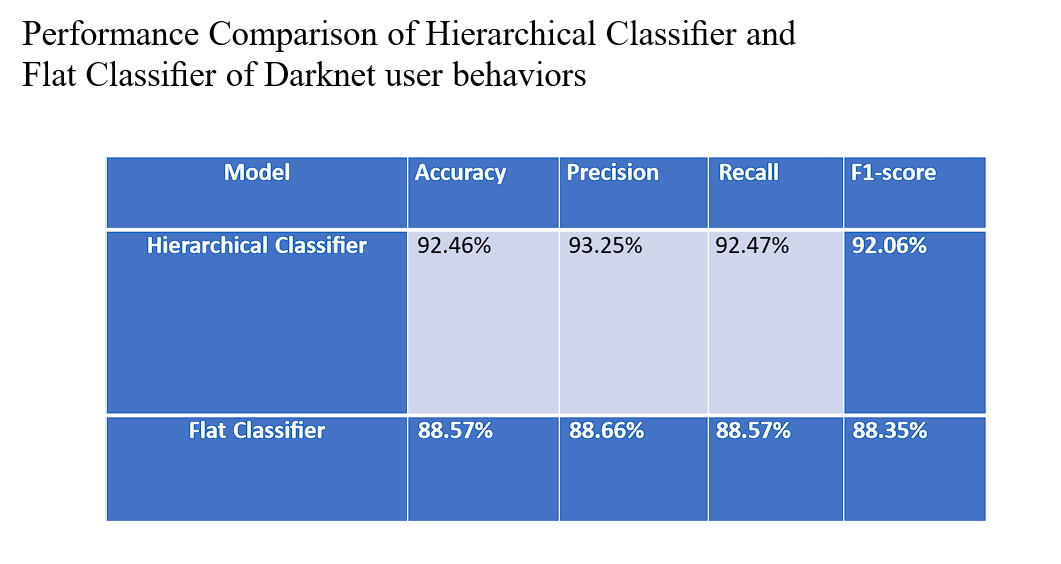
**First Layer:** Determines whether the traffic is from the darknet or normal internet.

**Second Layer:** Identifies the specific type of darknet (Tor, I2P, ZeroNet, or Freenet).

**Third Layer:** Determines the specific user behavior within the identified darknet.

This method allows for separate training of local classifiers for each parent node, enabling the selection of the most suitable classification algorithm for each specific task. However, a potential drawback is error propagation, where an error at one layer can affect subsequent layers' classifications​​.

Performance comparison of Hierarchical Classifier and Flat Classifier of Darknet user behaviors

 Fig.5.2 Performance comparison of Hierarchical Classifier and Flat Classifier of Darknet user behaviors

In the study of darknet user behavior classification, two primary approaches were compared: the hierarchical classifier and the flat classifier(Fig.5.2). The hierarchical classifier was designed to manage 25 different user behavior categories within darknet traffic. The flat classifier, specifically utilizing the XGBoost model, served as a benchmark due to its strong performance in this context.

The higher performance metrics of the hierarchical classifier indicate its greater effectiveness in identifying and categorizing darknet user behaviors. Furthermore, the confusion matrix analysis revealed that the hierarchical classifier had a significantly lower false prediction rate compared to the flat classifier. This suggests that the hierarchical classifier is more adept at minimizing error propagation, leading to more accurate and reliable classification results ​​.

Darknet Dataset at User Behavior Granularity

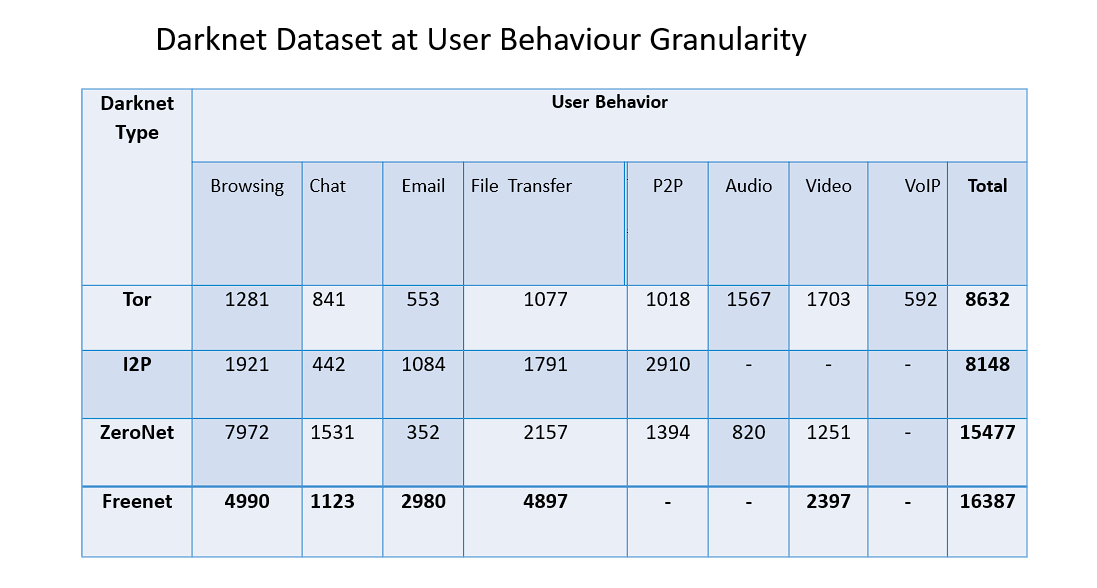


Fig.5.3 Darknet Dataset at User Behavior Granularity

The dataset used in this study included traffic data at a fine granularity level, focusing on user behaviors within various darknet environments, including Tor, I2P, ZeroNet, and Freenet(Fig.5.3). This granularity allowed for a more detailed and nuanced understanding of user activities, enhancing the ability to classify and differentiate between different types of darknet usage effectively. The dataset was meticulously labeled to reflect specific user behaviors, enabling the training of classifiers to achieve high accuracy in behavior identification .

Accuracy and F1 score of 6 local classifier:

To determine the most suitable classification algorithm for each local classifier within the hierarchical structure, six different local classifiers were evaluated (Fig 5.4): Darknet-Normal, Darknet-type, Tor-behavior, I2P-behavior, Freenet-behavior, and ZeroNet-behavior. The performance metrics used were accuracy and F1 score, with results indicating that machine learning methods, particularly XGBoost, generally outperformed deep learning methods in this context.

These results underscore the effectiveness of XGBoost and other machine learning methods in accurately classifying different types of darknet traffic based on user behaviors ​​.

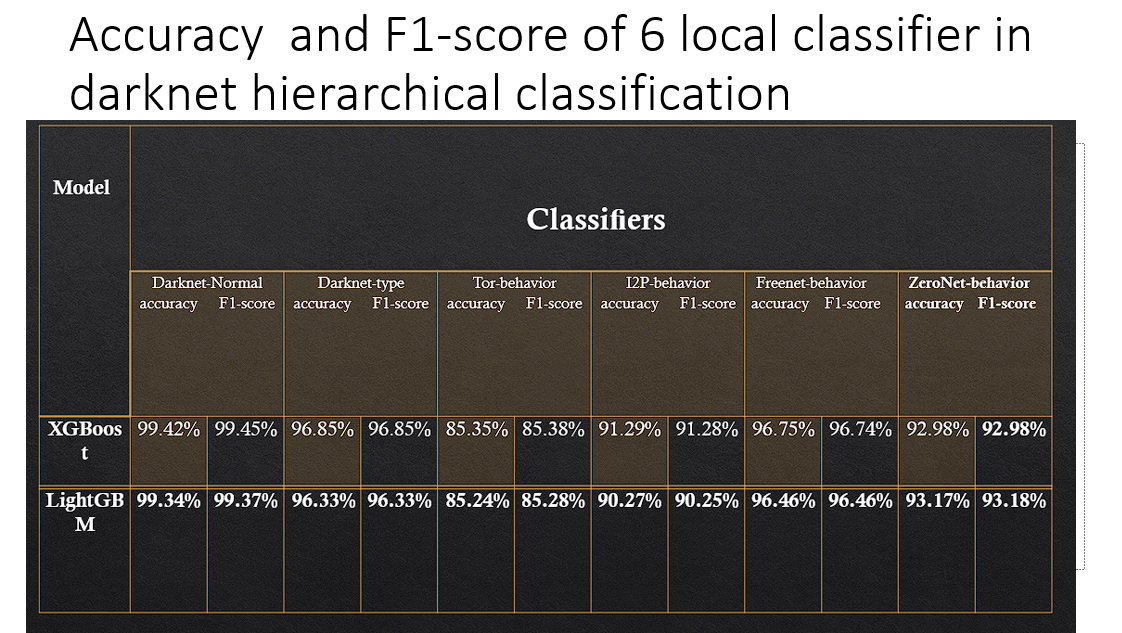


Fig.5.4 Accuracy and F1 score of 6 local classifier

Confusion matrix of 6 local classifier:

The confusion matrices for the six local classifiers provided insight into their classification performance and the types of errors made. Each matrix illustrated how well each classifier differentiated between the various user behaviors and the extent to which false positives and false negatives occurred.

For instance, the confusion matrix for the Darknet-Normal classifier using XGBoost showed minimal misclassifications, highlighting its robustness in distinguishing normal internet traffic from darknet traffic. Similar trends were observed in the other classifiers, with machine learning methods generally exhibiting higher precision and recall rates compared to deep learning methods.

Overall, the confusion matrices validated the high accuracy and reliability of the chosen algorithms, particularly XGBoost, in classifying darknet traffic based on user behaviors, thereby supporting the use of a hierarchical classification approach over a flat classification model .

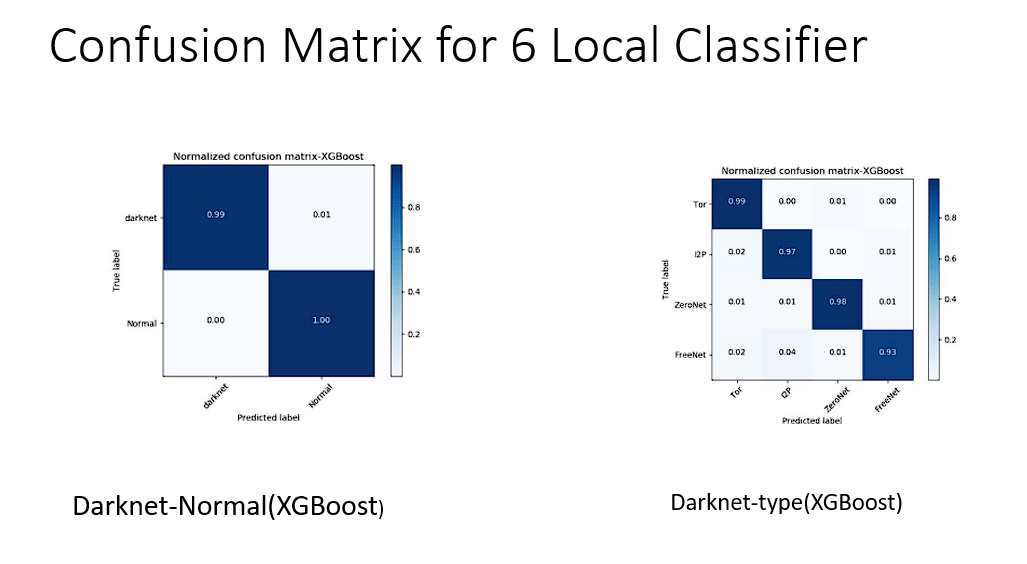


Fig.5.5 Darknet-Normal (XGBoost) Fig.5.6 Darknet-type(XGBoost)

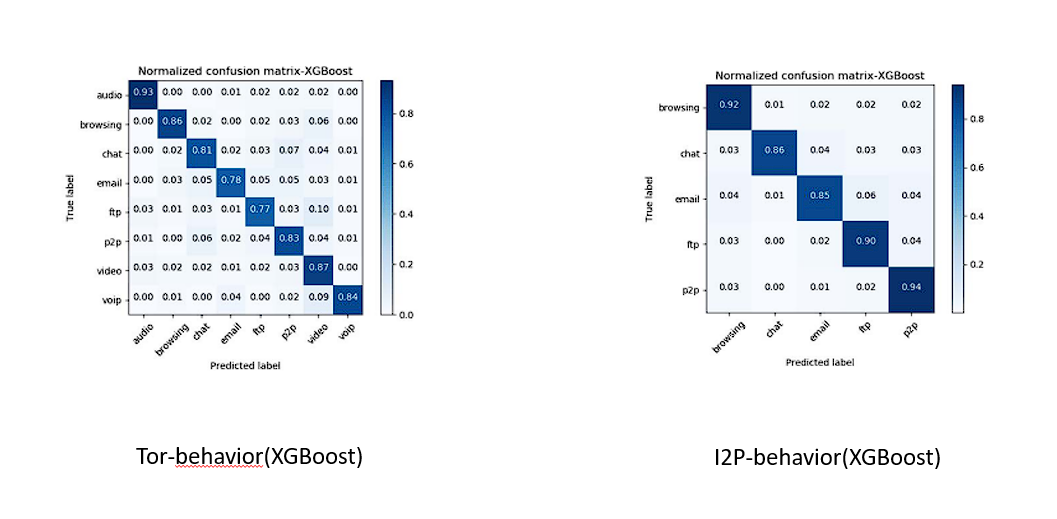


Fig.5.7 Tor-behavior(XGBoost) Fig.5.8 I2P-behavior(XGBoost)

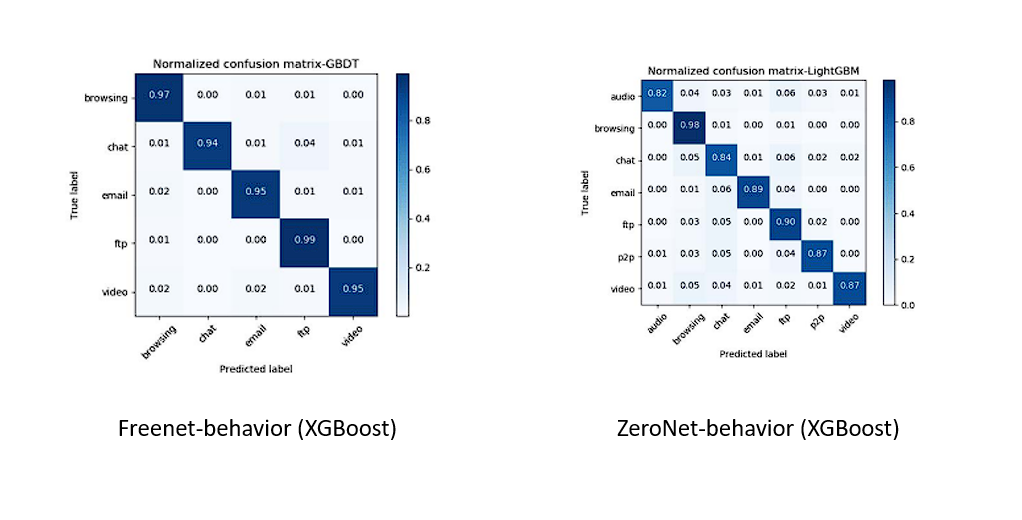


Fig.5.9 Freenet-behavior(XGBoost) Fig.5.10 ZeroNet-behavior(XGBoost)

**Risk Analysis**

The paper highlights the importance of darknet traffic classification to enhance security and user behavior traceability. Key points include:

**Traffic Classification:** Various machine learning techniques such as ANN, SVM, Naïve Bayes, Bayesian Network, C4.5, and Random Forest were utilized for classification. The hierarchical classifier proposed in the paper showed improved recognition accuracy (96.9% for darknet types and 92.46% for user behaviors) over flat classifiers​​.

**Dataset Generalization**: To improve model generalization, the dataset includes traffic under different network bandwidths, which can enhance model performance across diverse scenarios.

**Challenges in Darknet Research:** The paper notes the difficulty in obtaining datasets, as many existing datasets are kept secret, hindering further research. The authors addressed this by collecting and publishing a detailed darknet user behavior dataset​​.

This document provides detailed instructions and methodologies for simulating various user behaviors on multiple darknet platforms, essential for researchers aiming to study and classify darknet traffic effectively. Additionally, it discusses the challenges and potential solutions in enhancing darknet traffic analysis and security.

**CHAPTER 6**

**CONCLUSION**

This paper researches the traffic classification of user behavior in Tor, I2P, ZeroNet, and Freenet. According to our knowledge, there are few studies on the analysis of darknet user behaviors, especially for new types of darknets like ZeroNet and new applications in mature darknets. We propose a hierarchical classification method for the identification of darknet user behavior, which is more suitable in this darknet scenario than the flat classification method due to the large number of predicted categories. The experiment shows that this method can recognize 4 types of darknet with an accuracy of 96.9% and recognize 25 user behaviors with an accuracy of 92.46%.

Since previous research did not provide public traffic datasets on the granularity of user behavior, we deployed a traffic collection probe in a real darknet environment and published the dataset used in our experiment. In the experiment, we extracted 26 time-based flow features from traffic files to train 6 local classifiers that constitute the three-layer hierarchical classifier. The results also indicate that deep learning methods do not perform well when feature extraction accurately represents traffic characteristics.

In the future, we plan to further supplement the traffic under multiple network bandwidths to expand the darknet dataset, which can be useful for improving the model's generalization. We also aim to link it with IDS to provide a supervision method that detects and blocks dangerous behaviors on the darknet​​.

**REFERENCES**

[1] Wang M, Wang X, Shi J, et al. "Who are in the Darknet? Measurement and Analysis of Darknet Person Attributes," 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC). IEEE, 2018. 948-955.

[2] Fachkha C, Debbabi M. "Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization," IEEE Communications Surveys & Tutorials, 2015, 18(2). 1197-1227.

[3] Syverson P, Dingledine R, Mathewson N. "Tor: The second-generation onion router," Usenix Security, 2004. 303-320.

[4] The Invisible Internet Project (I2P). Jul. 2020. [Online]. Available: https://geti2p.net/en/

[5] ZeroNet. Jul. 2020. [Online]. Available: https://zeronet.io/

[6] Clarke I, Sandberg O, Wiley B, et al. "Freenet: A distributed anonymous information storage and retrieval system," Designing privacy enhancing technologies. Springer, Berlin, Heidelberg, 2001. 46-66.

[7] Cuzzocrea A, Martinelli F, Mercaldo F, et al. "Tor traffic analysis and detection via machine learning techniques," 2017 IEEE International Conference on Big Data (Big Data). IEEE, 2017. 4474-4480.

[8] Jansen R, Juarez M, Galvez R, et al. "Inside Job: Applying Traffic Analysis to Measure Tor from Within," NDSS, 2018.

[9] Berghel H. "The Deep Web: Beneath the Surface of the Internet," Communications of the ACM, April 2001.

[10] Xu X, Hua G, Xu K. "Characterizing User Behavior in Online Social Networks," 2010 IEEE Second International Conference on Social Computing. IEEE, 2010.