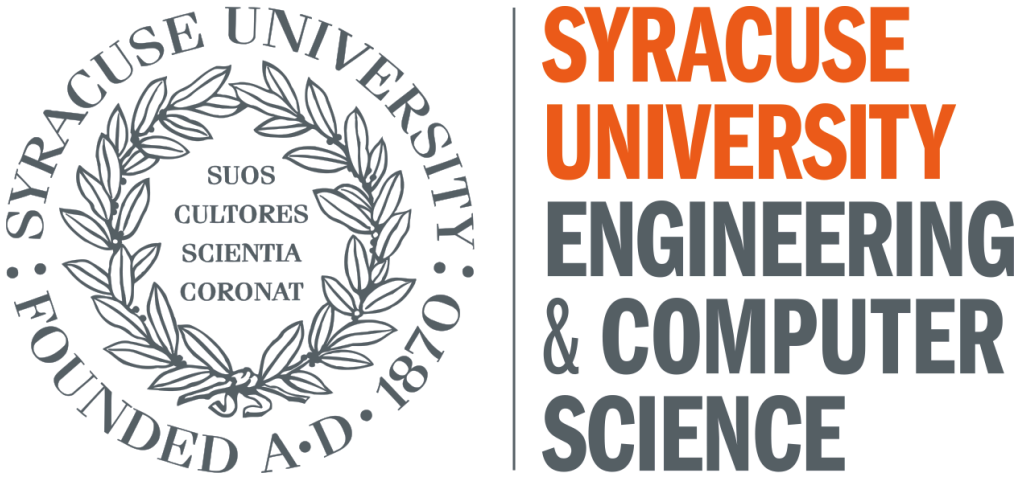
CIS600 Applied Natural Language Processing

Spring 2024 Term Research Project Proposal

"Malicious URL's detection using NLP and ML"

Team Information

Prasanth Sagar Kottakota	(SUID: 969978673)
Sri Krishna Chaitanya Chivatam	(SUID: 290797110)
Shyam Sudheer Nadella	(SUID: 364923152)
Vaibhav Krishna Joshi	(SUID: 556679674)
Uday Kumar Putta	(SUID: 317626035)



1. Project Idea:

In today's digital world, cyberattacks are increasing, posing serious threats to user security. This project offers a proactive defense by predicting the safety of URLs. Users input a URL, and the system quickly evaluates its safety, acting as a gatekeeper before access. By using machine learning models and NLP techniques, the system identifies potentially harmful URLs, enhancing online safety and protecting users from cyber threats.

Project Features:

- **URL Input**: Users can input any URL they intend to visit into the system.
- **Data Preprocessing** : Every URL data is more important, But when converting it into token, we are looking to use some preprocessing techniques like removing slashes, https:// and numerical information which might not be of any use in some cases. We are still researching URL properties.
- **Feature Engineering and Natural Language Processing**: The URL's are text data, we are looking to do feature engineering and work on creating more attributes specific to URL like length, URL specifications, Google index , TLD etc.. Similarly 22+ other attributes, But not only this, we are looking to use the URL as well, we would convert it into a tokens using BERT[1] tokenizer and create embedding as well which will capture the contextual information of the URL's allowing the models to understand the semantic meaning.

  Though BERT offers a better way of semantic understanding, we would like to explore some other text processing and feature understanding techniques commonly used in NLP, such as TF-IDF, Countvectorizer , one hot encoding and hashing as well.

- **Machine Learning Model**: Utilizing machine learning algorithms, the system will analyze the characteristics of the entered URLs to determine their safety status.
- **Safety Assessment**: Based on the analysis, the system will classify the URLs into two categories: safe or malicious. Users will receive immediate feedback regarding the safety status of the URL they provided.
- **Deployment**: Our application transcends development; it's deployed as a website using the Django framework. This seamless integration ensures a smooth user experience, allowing input via the UI, backend analysis, and result delivery through the interface.
- **Continuous Learning**: Our system evolves with user inputs. It retains URLs for ongoing learning, periodically updating the model based on confirmed URL authenticity by admin.

---

[1] BERT (Bidirectional Encoder Representations from Transformers) is considered a pre-trained model because it is initially trained on a large corpus of text data in an unsupervised manner before being fine-tuned on specific downstream tasks.

Techniques and Algorithms To be Used:
Tokenization (For TF-IDF, Count vectorizer etc..)
Lemmatization
Hashing Vectorizer
TF-IDF
Countvectorizer
NLP Model (BERT) for tokenization and embedding generation
Classification Algorithms(Random forest, Logistic etc..) But not limited to..

2.Significance of the Idea:

Our project helps protect people from online dangers by stopping harmful websites before they cause harm. Instead of just reacting after a problem occurs, our system acts beforehand, giving users the power to avoid risky websites.

We use ML algorithms to analyze websites and decide if they're safe or not. This way, our system can quickly adapt to new threats, keeping users safe from the latest cyber dangers. By using this approach, we're giving people a better way to stay safe online, using the latest technology to stay one step ahead of cyber threats.

The way, I consider this project brings an importance into online security, it would help cyber security companies or firewalls to detect the malicious URL's before hand increasing a better way into security and growing web development, and that would boost the performance of a software and better ways of predicting malicious url before hand using NLP.

3.Work Plan and Tasks:
The work plan includes research paper reading, data collection, pre processing, feature engineering, model development, Deployment with Django and documentation

Data Collection and Preprocessing: We'll gather data from various online sources like Kaggle, GitHub, and research labs. After collecting, we'll organize it into dataframes, labeling, removing duplicates, special symbols, generating tokens, and fixing any missing values and other data preprocessing tasks mentioned in project features.
Feature Engineering: In addition to URLs, we'll include features like URL length, domain names, abnormality, suspicious words, IP address count etc.. in our training dataset. These features will help our model understand and classify URLs better.
Natural Language Processing: The feature engineering focuses on finding more relevant attributes or specifications regarding URL, But the NLP in our project focuses on understanding semantics of URL, we tokenize and generate embeddings using BERT and then feed this data along with some other features into the ML model. This would help the model to classify the URL's better and more meaningful. And apart from the BERT , we also use some other traditional NLP techniques to convert text into features such as TF-IDF, Count vectorizer etc..

Research and Model Development: We'll start by studying existing research to find the best classification methods. Then, we'll build and test our model using these methods or algorithms to ensure it works well. As far as our existing plan For classification, we'll consider models like Random Forest and Logistic Regression, and possibly others. We'll choose the ones that work best for our needs.

Deployment in Django: Using our experience with Django, we'll create a user-friendly UI and integrate our model into the backend. This will allow users to easily input URLs and receive safety predictions.

Documentation: We'll thoroughly document the entire process, including successful approaches and any challenges faced. This documentation will serve as a valuable resource for future reference and improvement.

Timeline:

Data Collection and Preprocessing	:- 4/1/2024 - 4/4/2024
Feature Engineering and NLP	:- 4/5/2024 - 4/11/2024
Research and Model Development	:- 4/15/2024 - 4/19/2024
Deployment in Django	:- 4/20/2024 - 4/22/2024
Documentation :	:- 4/25/2024 - 5/2/2024

Task Distribution:

Data Collection and Preprocessing	- Uday Putta, Vaibhav Joshi
Feature Engineering and NLP	- Sri Krishna Chaitanya Chivatam, Prasanth Sagar
Research and Model Development	- Shyam Sudheer, Prasanth Sagar
Deployment in Django	- Uday Putta, Vaibhav Joshi
Documentation :	- All the team members

References :

https://ar5iv.labs.arxiv.org/html/2310.05953
https://ieeexplore.ieee.org/abstract/document/9633889