Supervised Approaches to Cybersecurity News Classification: Classical ML and Transformer Models

CHRISTOPHER ORELLANA, MAUNG AUNG, MARY UCHE ONONYE, JUDY SUPERSAD, VIKRANT SHARMA

Introduction and Background

Why are we analyzing the dataset?

- We are analyzing this dataset to understand how we can identify different cyber threats
- Studying these articles helps us see patterns that indicate malware, data breaches and other attacks

What is the purpose of our Analysis?

- The purpose of our analysis identify meaningful linguistic patterns
- Examine word usage, frequent phrases, grammatical structures and build out robust models to predict different type of attacks

What's our plan?

- Clean and preprocess the text, analyze patterns using tokens, lemmas, bigrams, trigrams and noun phrases
- Use the insights to engineer strong features for machine learning models to classify type of cyber attacks

Motivation of research

- How can automated systems, like those used by cybersecurity teams, analyze text, detect patterns, and classify potential threats before they reach employees?

- How effectively can a fine-tuned transformer model, optimized with hyperparameter, classify real-world cybersecurity news articles into threat categories, and what performance pros/cons does this automated approach offer over classical ML methods?

Overview of the dataset

Basic Overview of our dataset

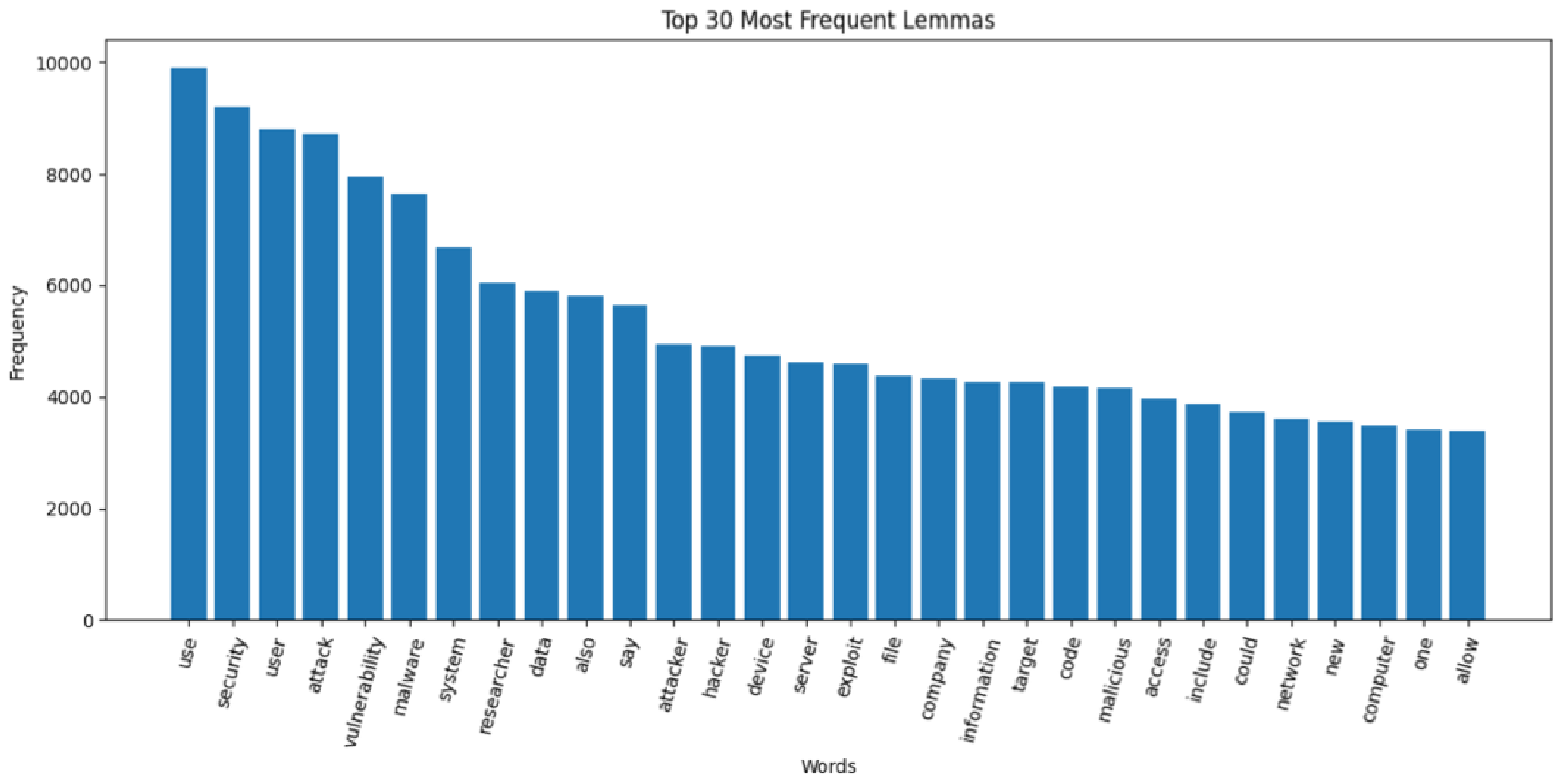
- 3,742 cybersecurity articles
- 4 Columns: x = Title, Link, Article
- y = Label

Start looking for patterns

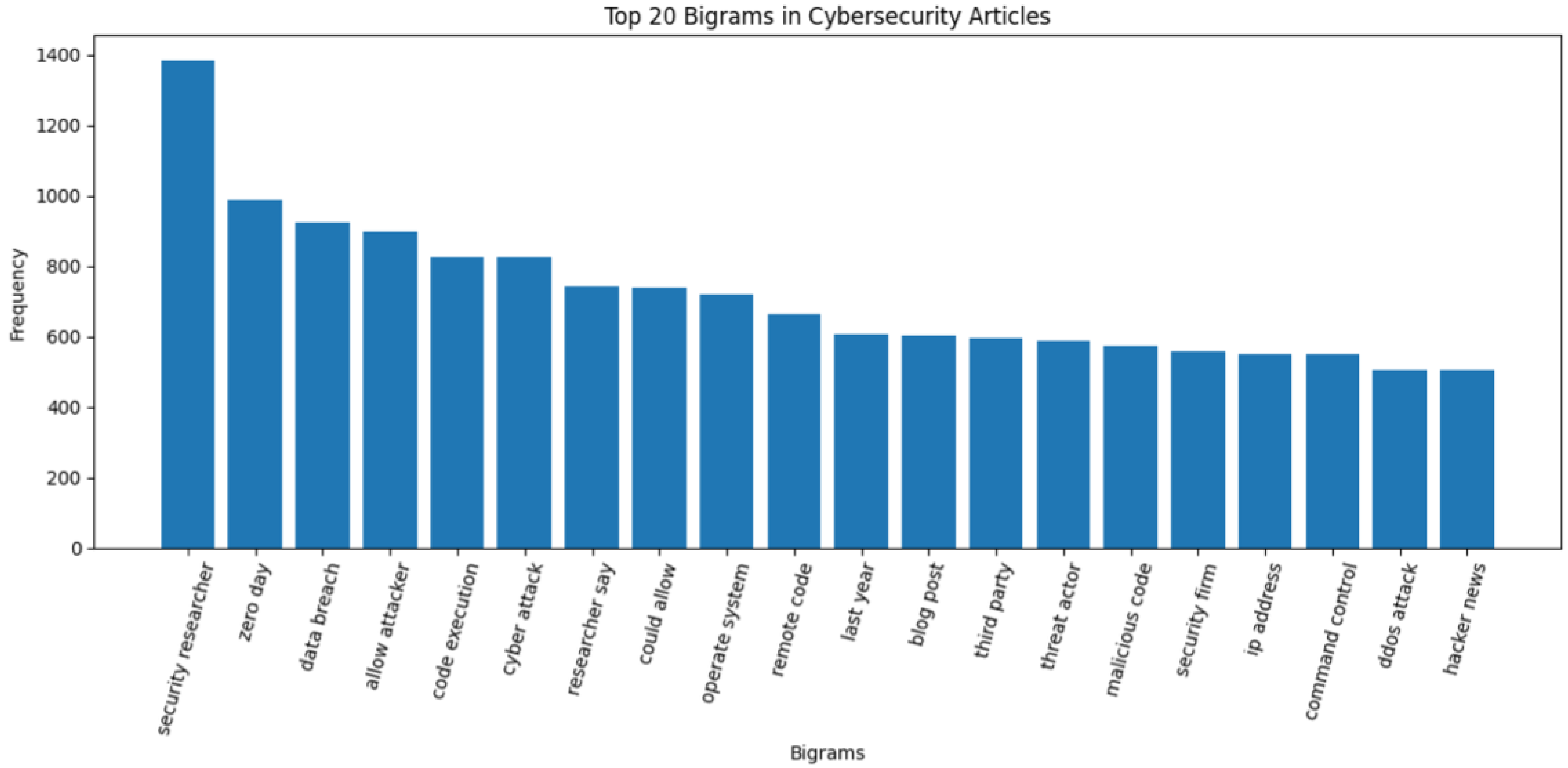
- Frequent bigrams/trigrams plotted
- Measured sentences and words

Variation in article length

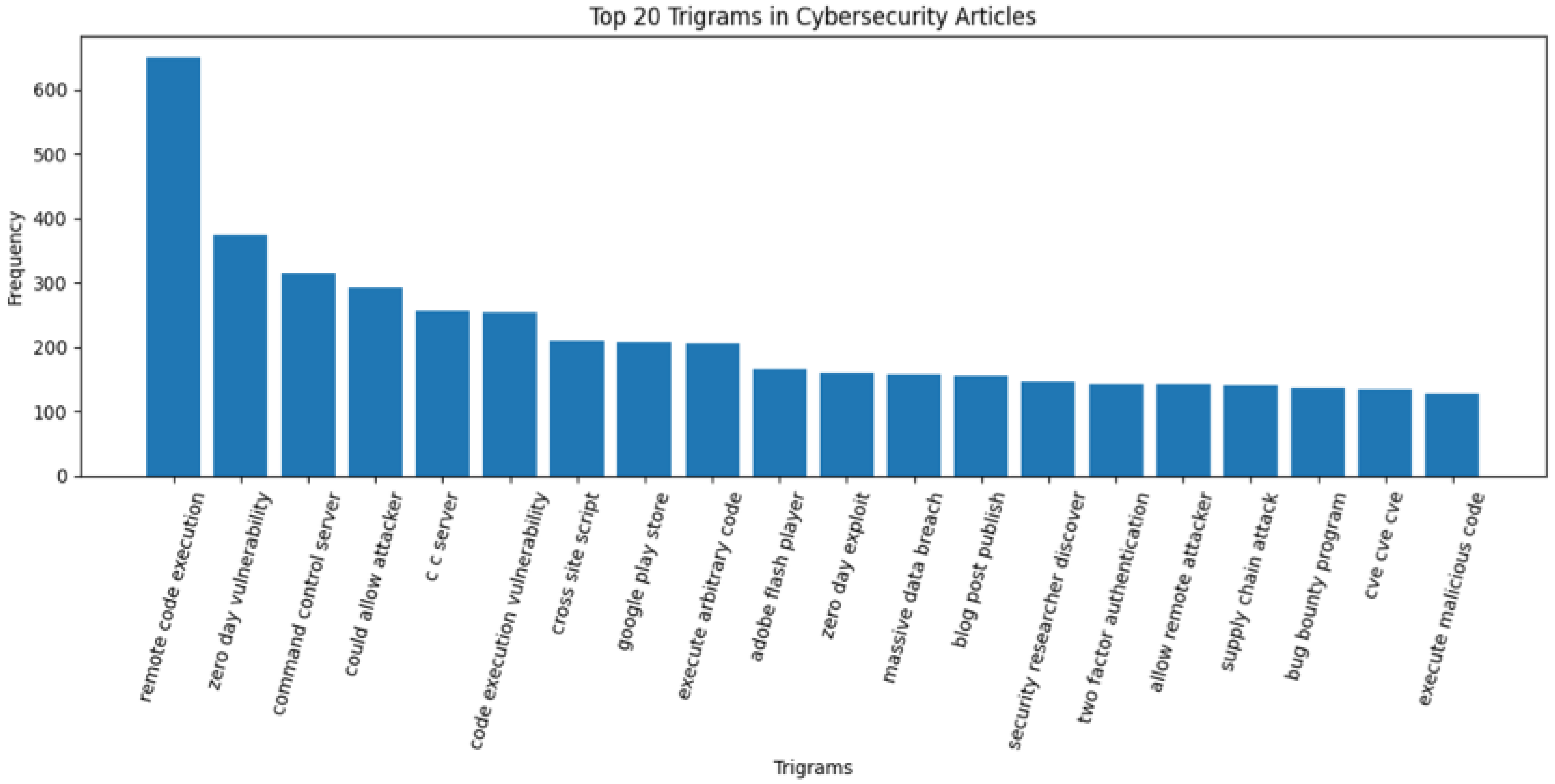
- 3,349 Long articles tend to investigative or deep dive
- 361 Shortest articles are breaking news
- 32 Short alerts



Top 20 Bigrams



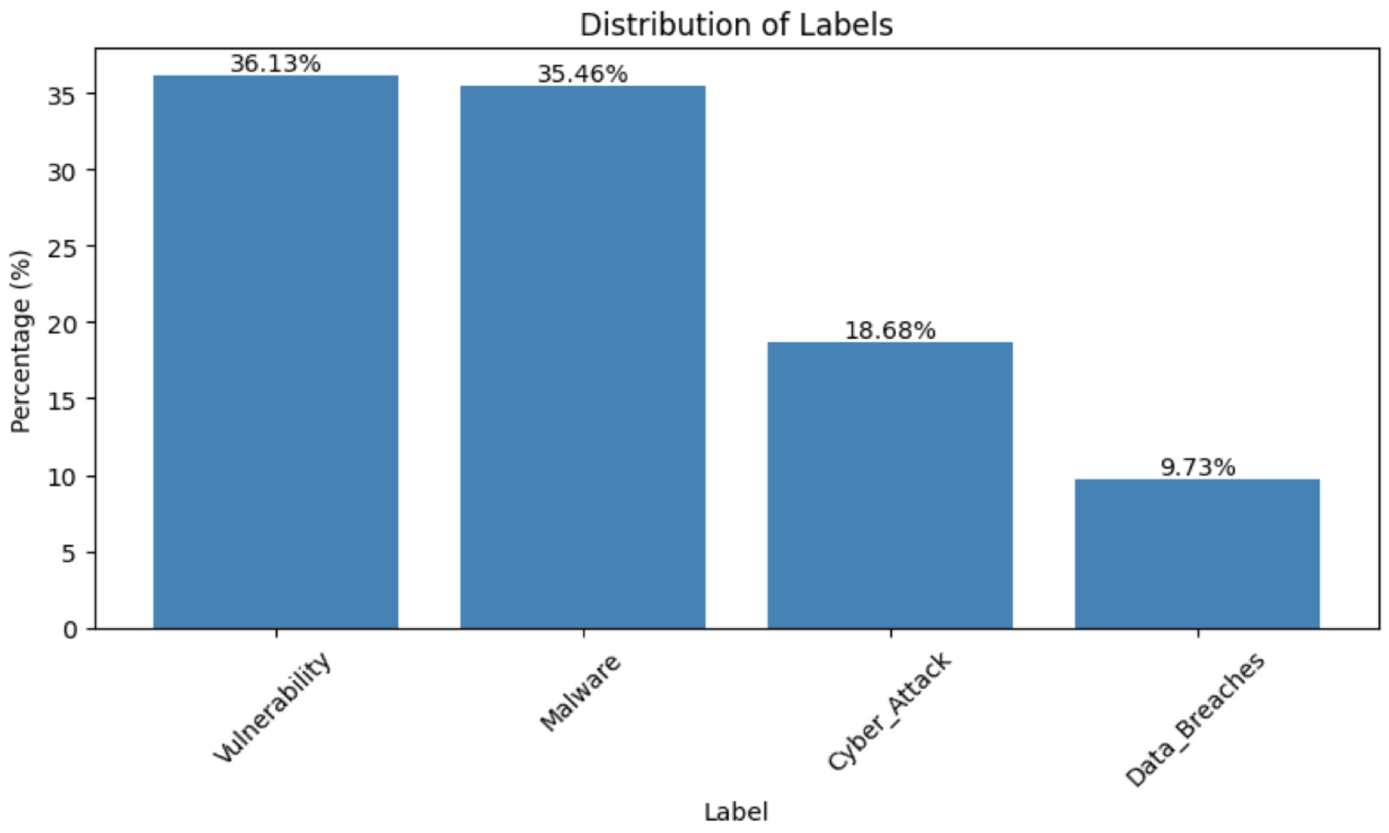
Top 20 Trigrams



Label Proportion Percentages

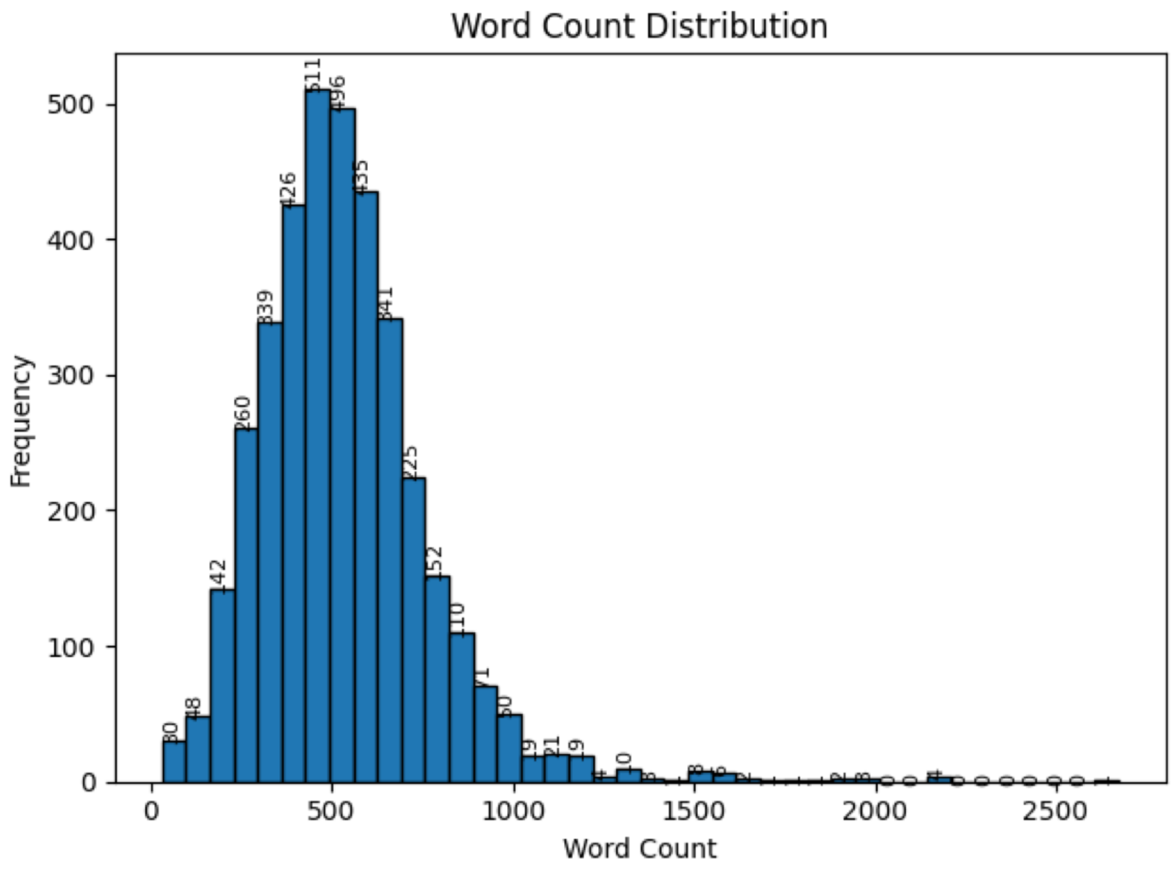
Vulnerability	36.13%
Malware	35.46%
Cyber_Attack	18.68%
Data_Breaches	9.73%

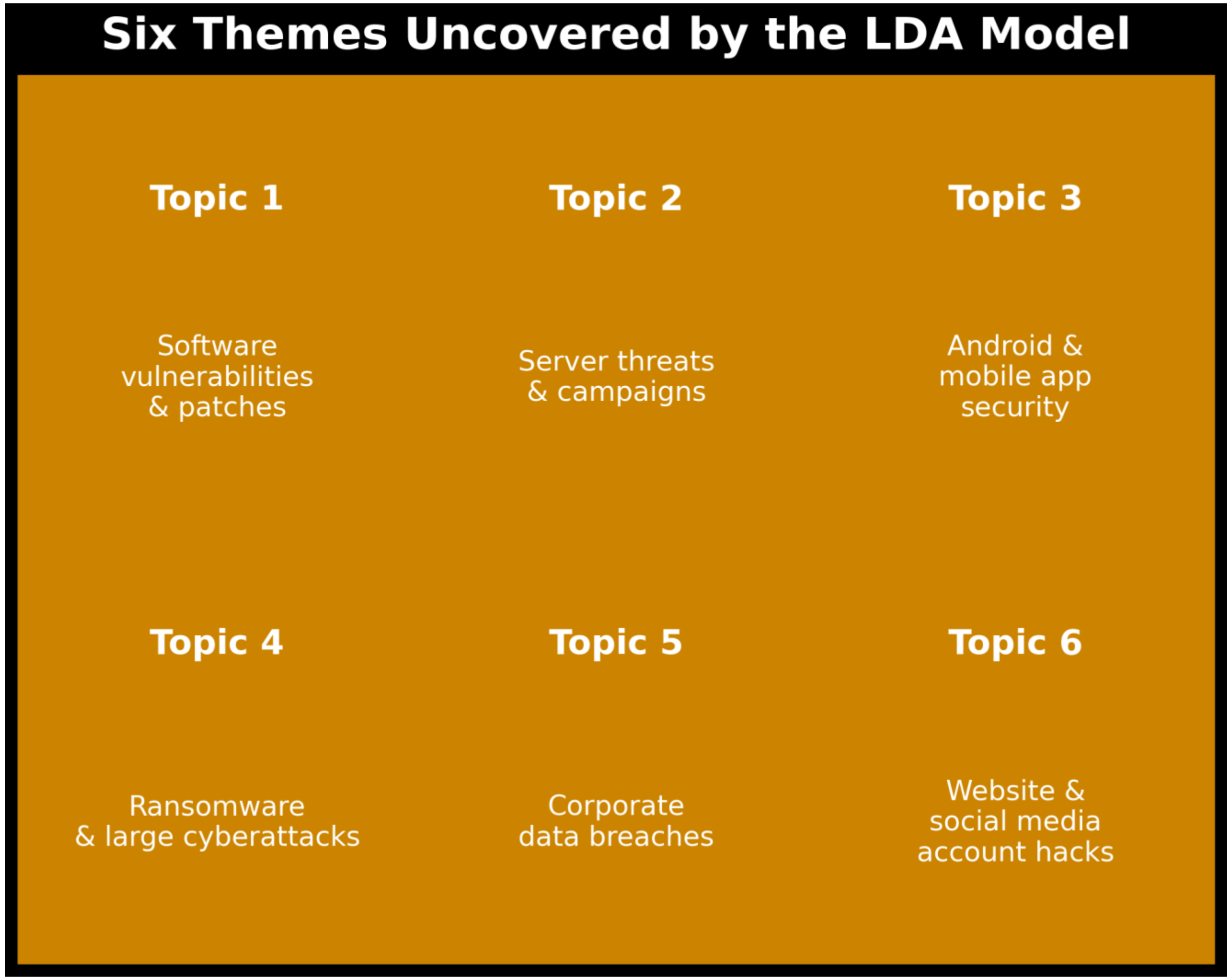
We can see there is a class imbalance for this dataset



Word Count Distribution

- Most articles fall in 300–700-word range
- Steep drop-off for very short or very long pieces
- Dataset is dominated by medium-length cybersecurity reports





Six Themes Uncovered by the LDA Model

Topic 1	Topic 2	Topic 3
Software vulnerabilities & patches	Server threats & campaigns	Android & mobile app security
Topic 4	**Topic 5**	**Topic 6**
Ransomware & large cyberattacks	Corporate data breaches	Website & social media account hacks

LDA Topic Modeling Setup

Goal

-Use LDA to uncover latent topics and compare them to original labels

Labels

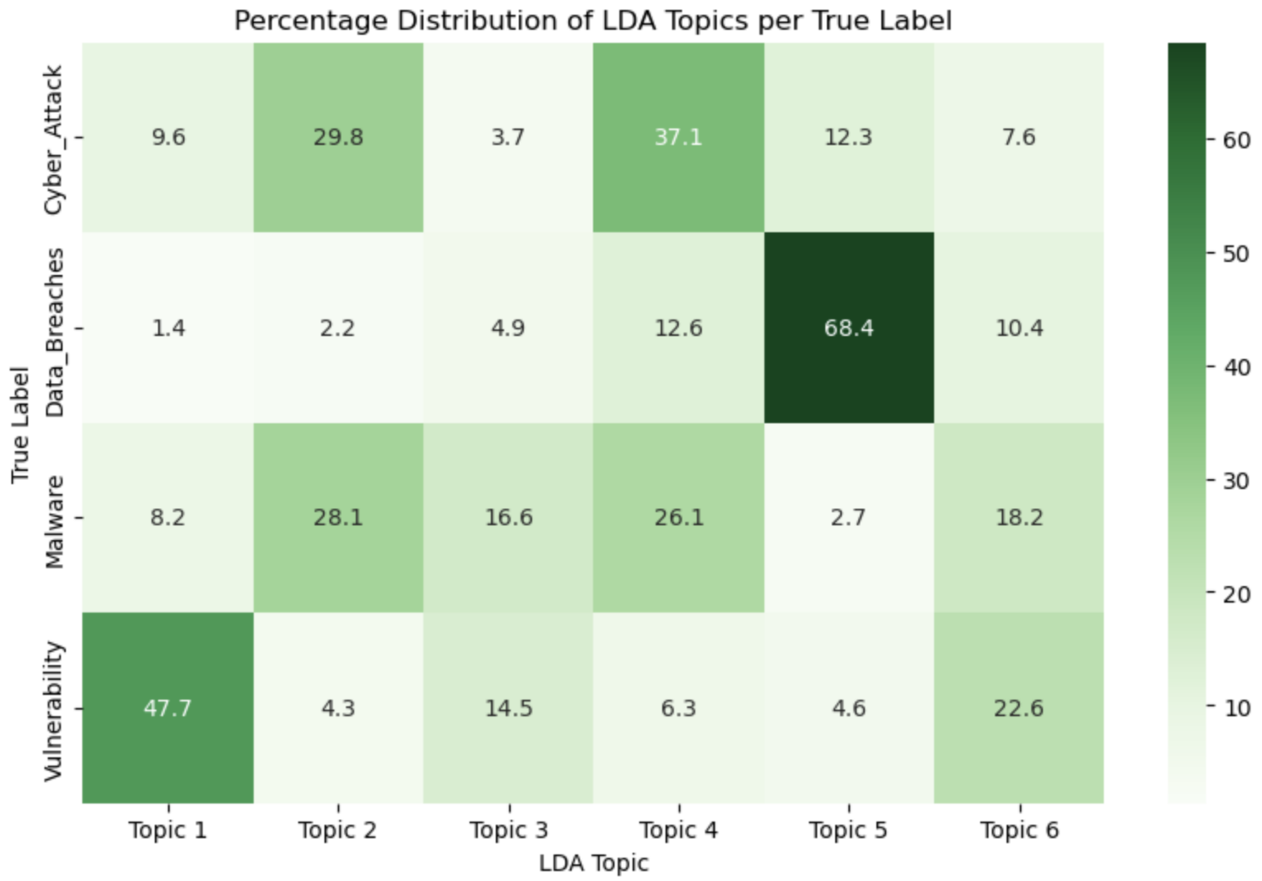
-Cyber_Attack, Data_Breaches, Malware, Vulnerability

Preprocessing

- Cleaned the text using spaCy: lowered the text, removed stopwords, lemmatized words
- Removed punctuation and very short/irrelevant words

LDA Model

- Model learned **6 topics** from the articles
- About **6,500 meaningful words** used after cleaning
- Trained over **10 passes** for stable results



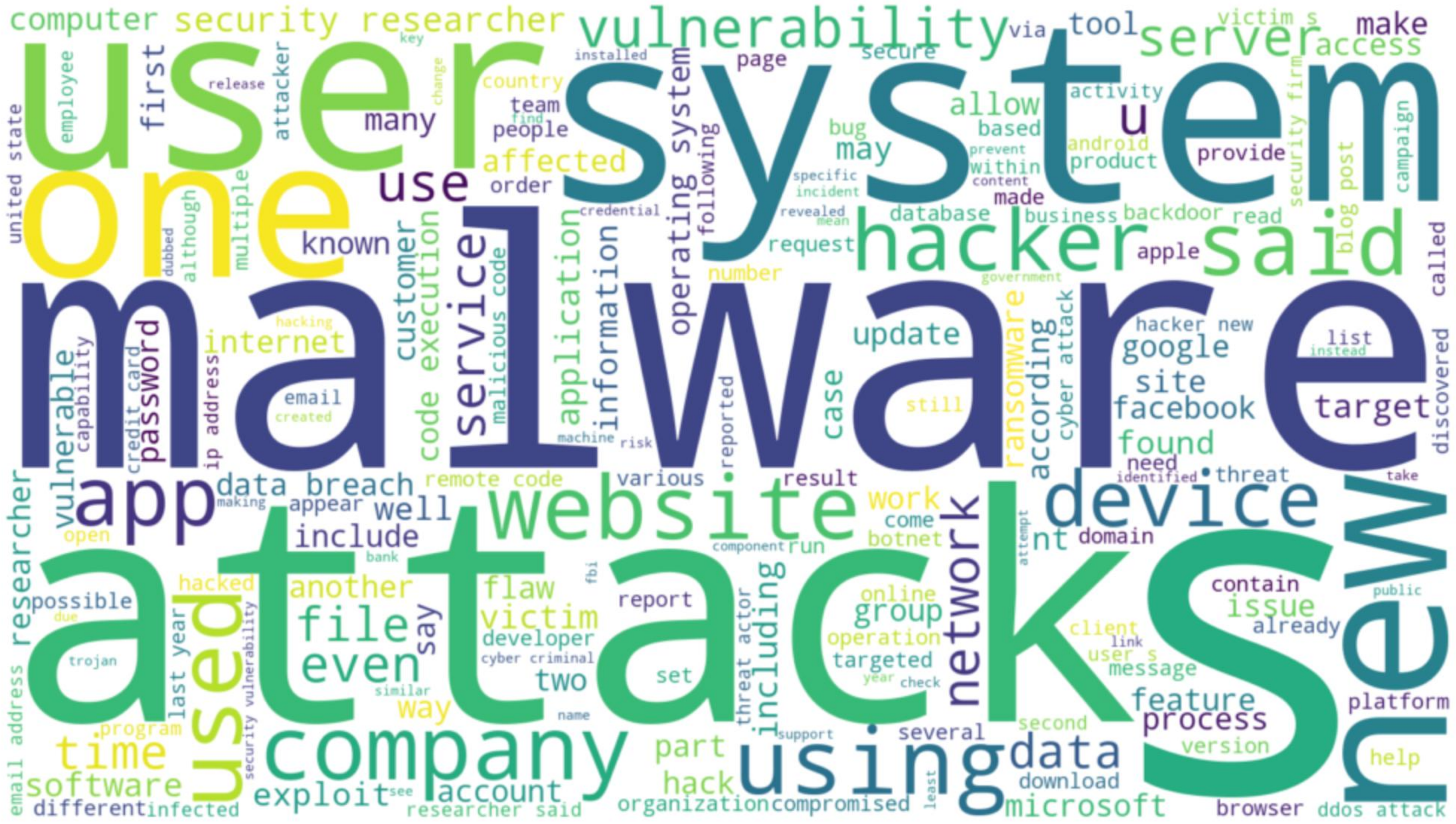
Comparing LDA Topics To Original Labels

- **Data Breaches** shows the strongest alignment, most articles fall into one clear topic.
- **Vulnerability** is fairly consistent but more mixed across several topics.
- **Cyber Attack** splits mainly between two themes, showing broader coverage.
- **Malware** is the most scattered label, appearing across multiple topics.
- These patterns show which labels are clearer vs. more overlapping in meaning.

Supervised Classification - Text Pre-processing

- Data Cleaning and Normalization -
  - Drop rows with missing article/label
  - Combined **Title + Article** into a single text field
  - Removed URLs, stopwords, punctuation
- Tokenization & Vocabulary Construction -
  - Tokenized text using NLTK
  - Applied **lemmatization** (using WordNetLemmatizer) to normalize word forms
  - Extracted top **1,500 informative unigrams**
  - Added **300 bigrams** to capture cybersecurity phrases (e.g., *zero day*, *data breach, etc*)
- Feature Engineering -
  - **Frequency distribution** over all tokens and kept the top 3000 most common and dropped generic ones like "security", "user", "system", "data", etc
  - Created **Indicator of Compromise (IOC)** regexes boolean flags like has_cve, has_ip, has_email, etc to capture any digital evidence of an attack (e.g., CVE-2023-12345, 192.168.0.1)
  - Defined **domain-specific keyword** sets for ransomware, malware, phishing, etc ,and created boolean flags to detect them.
  - Created **POS buckets** (many_proper_nouns, verb_heavy, noun_heavy)

Most Common terms



Text Analytic Methods

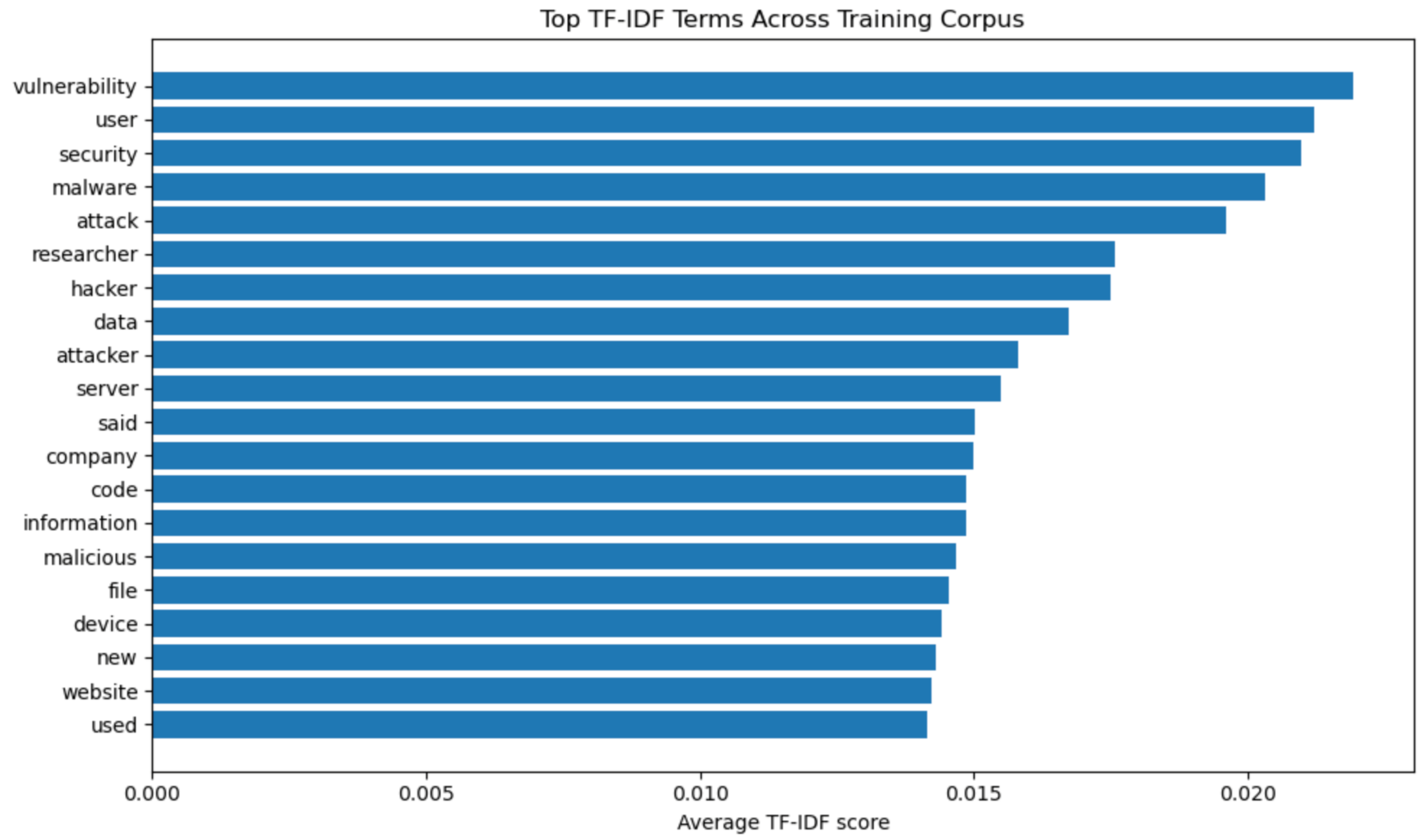
NLTK classifiers
- Naïve Bayes
- Decision Tree
- MaxEnt GIS

Sklearn classifiers
- Logistic Regression
- Randon Forest
- Gradient Boosting
- KNN (k=5)

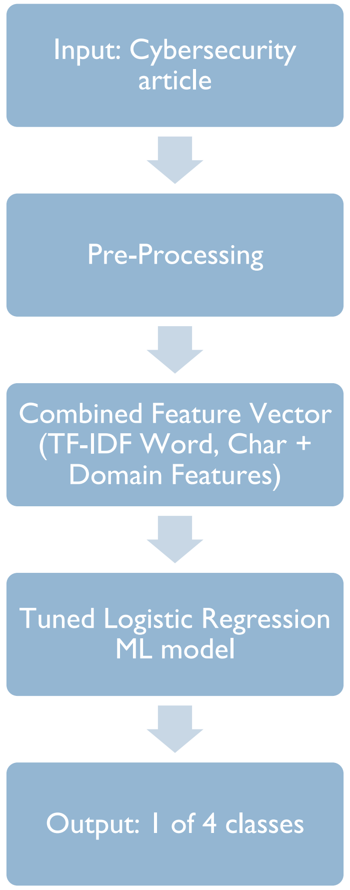
Hyper-Parameter Tuned
- Logistic Regression, 5-fold, scoring = "f1-macro"

Text Analytic methods:
- **Word-level TF-IDF**: n-grams = (1, 3), min_df=3, max_features=30,000
- **Character-level TF-IDF**: char n-grams = (3, 6)
- Concatenated as [word_TF-IDF | char_TF-IDF]
- Appended **domain features**:
  - Keyword flags: has_ransom, has_phish, has_vuln, etc
  - IOC flags: has_cve, has_ip, has_hash, has_email
  - Structural & POS buckets

TF-IDF terms


Top TF-IDF Terms Across Training Corpus
vulnerability
user
security
malware
attack
researcher
hacker
data
attacker
server
said
company
code
information
malicious
file
device
new
website
used
0.000
0.005
0.010
0.015
0.020
Average TF-IDF score

Working of Model

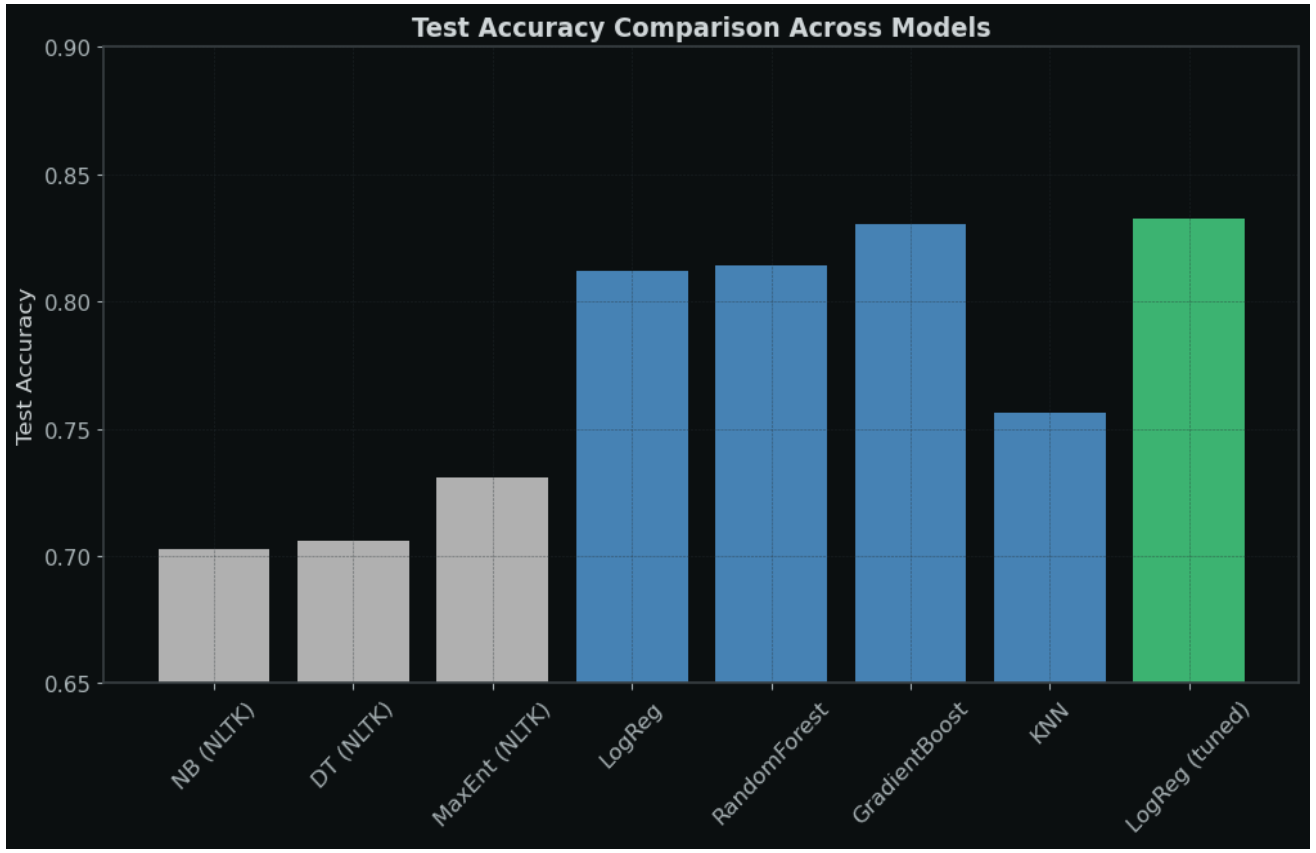

Input: Cybersecurity article
Pre-Processing
Combined Feature Vector (TF-IDF Word, Char + Domain Features)
Tuned Logistic Regression ML model
Output: 1 of 4 classes

Why this approach?

- TF-IDF alone misclassifies **Cyber Attack** and **Vulnerability** due to shared vocabulary
- IOC patterns (CVE, IP, hashes) give strong signals for **Vulnerability & Malware**
- Character n-grams capture technical identifiers TF-IDF misses
- Domain keywords help disambiguate subtle differences between threat types
- POS + structure features separate narrative news from attack writeups

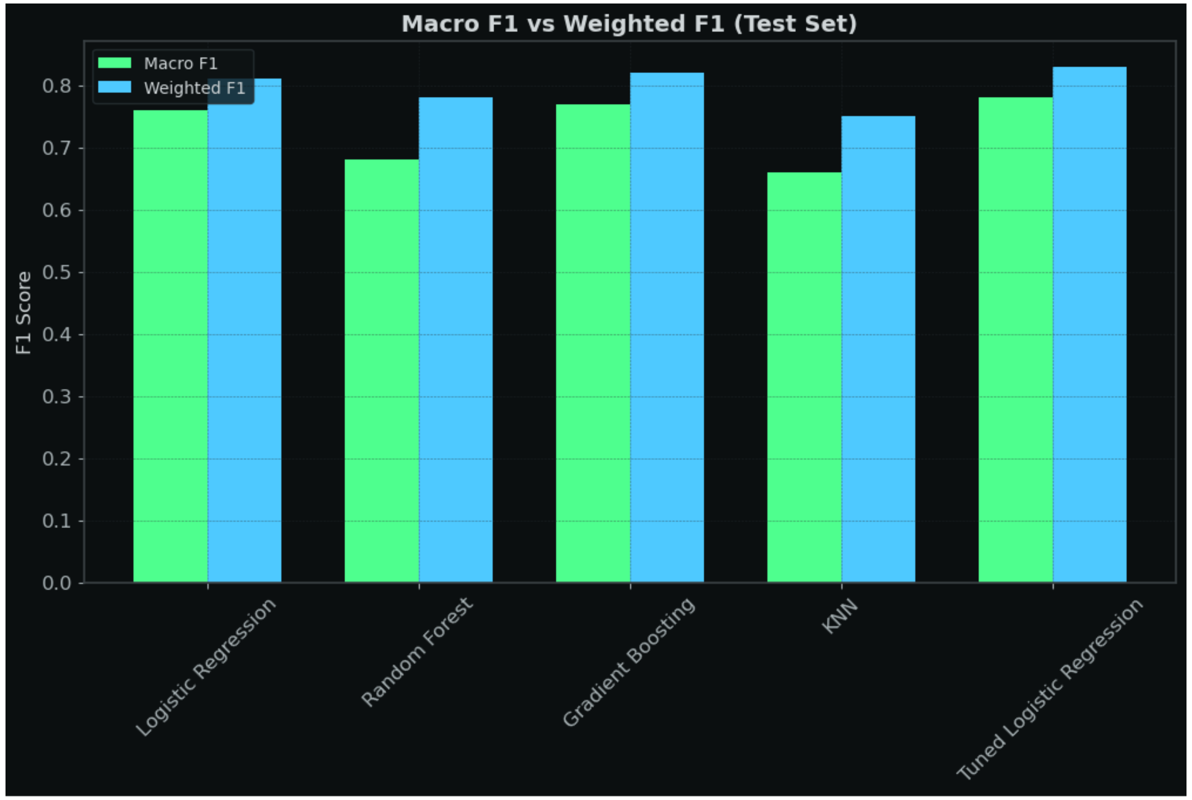
Model Performance Comparison (Supervised Learning Methods)

- NLTK models (NB/DT/MaxEnt) → 70–73%
- TF-IDF models (LogReg/RF/GB/KNN) → 81–83%
- Tuned Logistic Regression = B**est model** → 83%
- TF-IDF features outperform handcrafted features
- More expressive text representation → higher accuracy



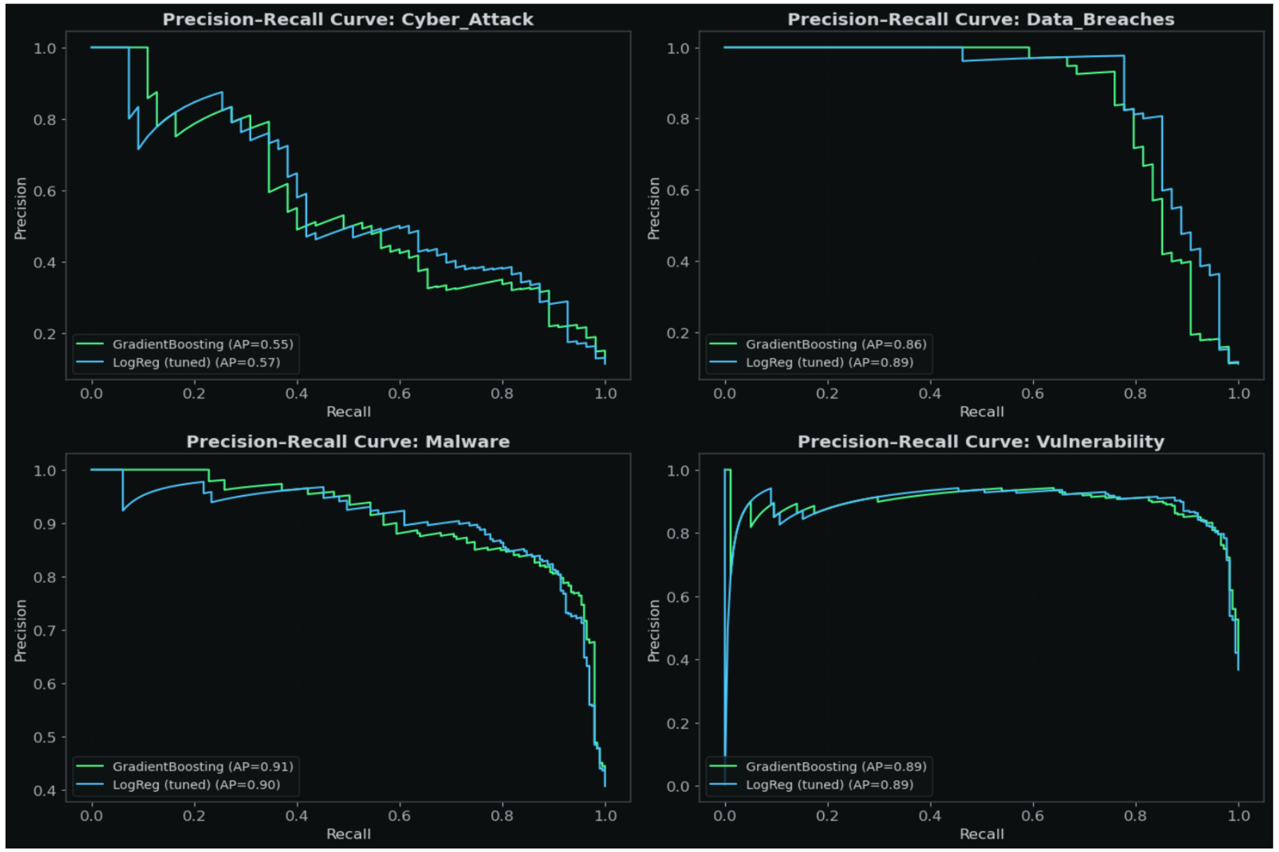
F1 Score Comparison Across TF-IDF Models

- Macro F1 → equal class weight
- Weighted F1 → adjusted for class sizes
- TF-IDF models show strong F1 scores overall
- Random Forest & KNN weaker on minority classes
- Tuned Logistic Regression = best-balanced model



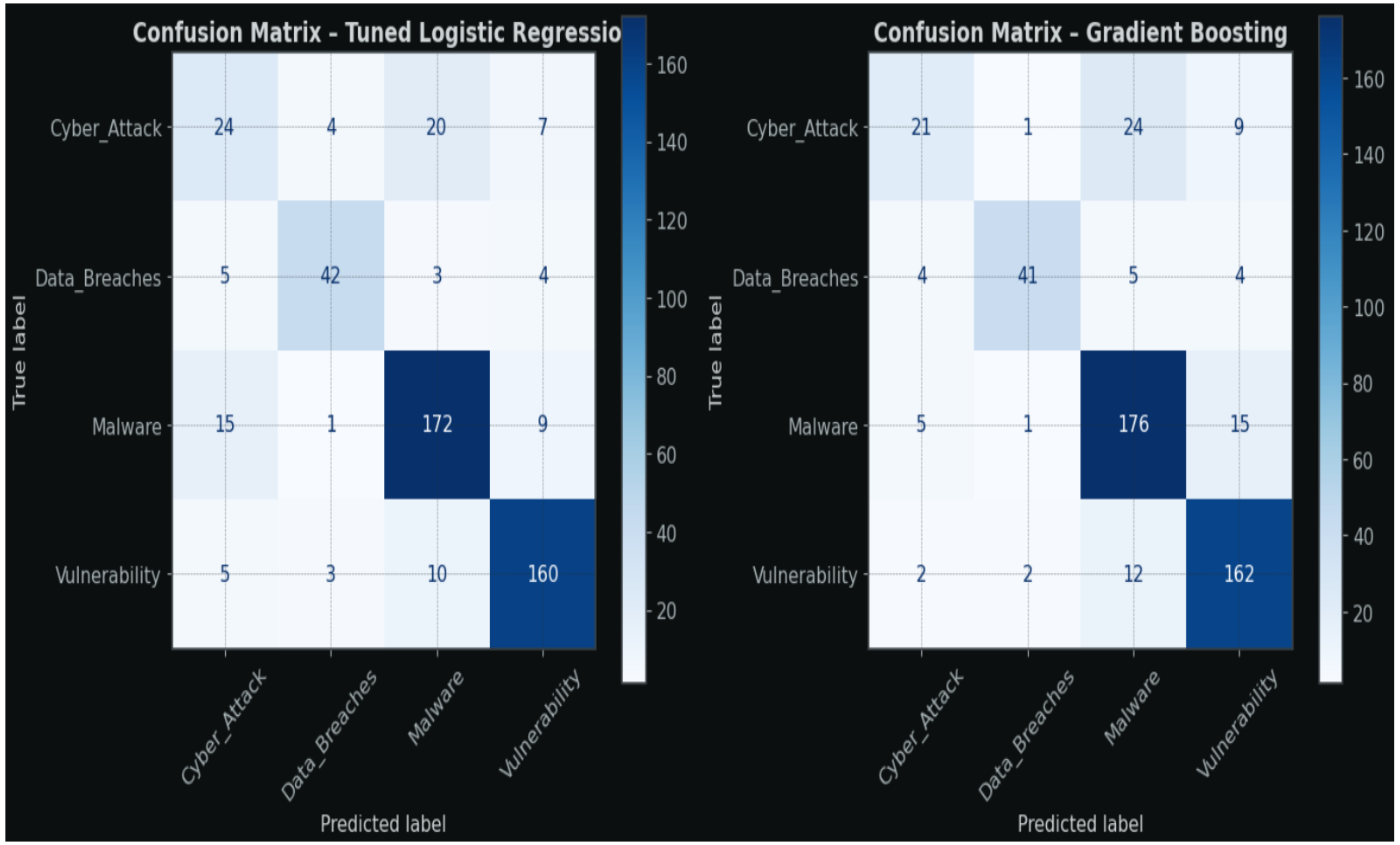
Precision–Recall Curves: Top Models by Class (Tuned LR vs GB)

- PR curves show precision–recall tradeoffs or each class
- Cyber Attack: LR slightly better
- Data Breaches: LR leads with higher AP
- Malware: GB slightly stronger
- Vulnerability: both models perform similarly
- Confirms Tuned LR is the most **consistent and balanced** model across all categories



Confusion Matrix Comparison – Tuned LR vs GB

- Shows where each model succeeds or struggles
- Tuned LR has a cleaner diagonal → fewer mistakes
- GB performs slightly better on Cyber Attack but makes more mistakes on Malware/Vulnerability
- LR shows more balanced, consistent across all classes
- Confirms LR Tuned as the most stable overall model



Text Classification using DeBERTa-v3-base

As a comparison we also ran a deep learning model: DeBERTa V3 Base - a powerful natural language understanding (NLU) model developed by Microsoft for Text Classification of different cyber security articles into cyber threat categories.

Preprocessing for DeBERTa v3

- Label encoder for huggingface model
- Label mapping saved in a JSON file
- 80% Training – 10% Validation, 10% Test splits
- DeBERTa tokenizer

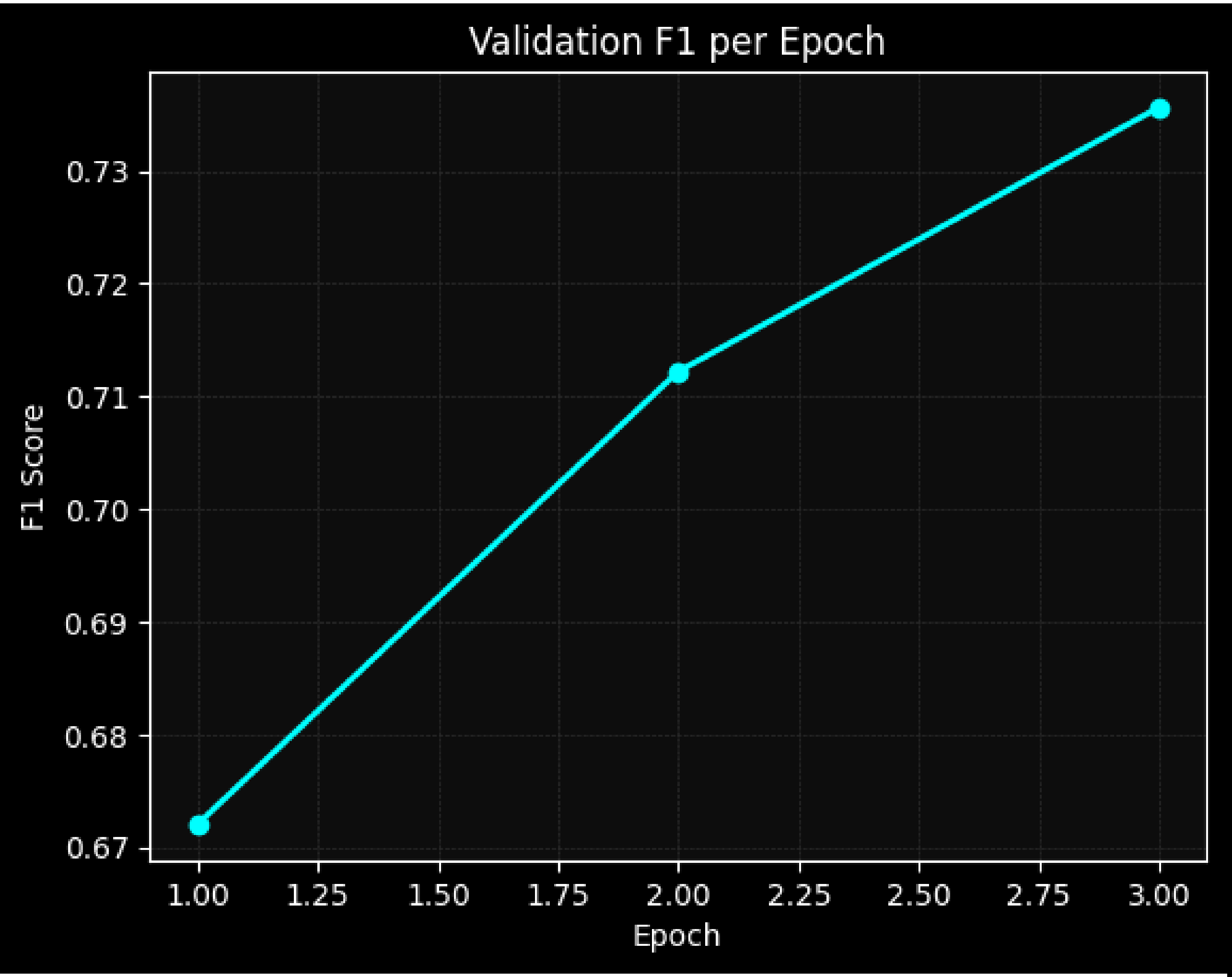
Fighting Class Imbalances

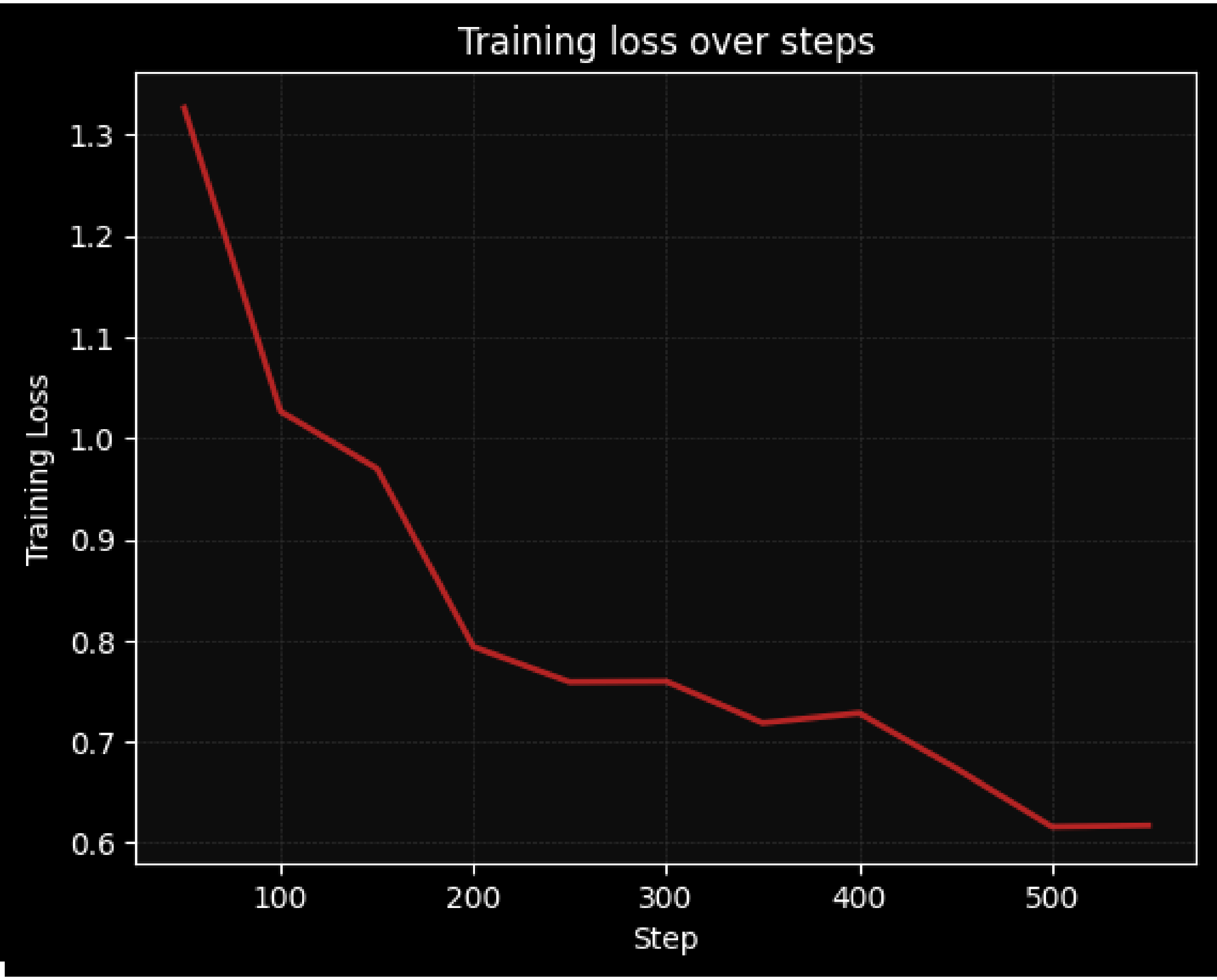
Since our dataset is imbalanced, when running the model, it tends to overclassify the most likely groups and misclassify the under-represented groups. We remedy this in 2 ways for this model:

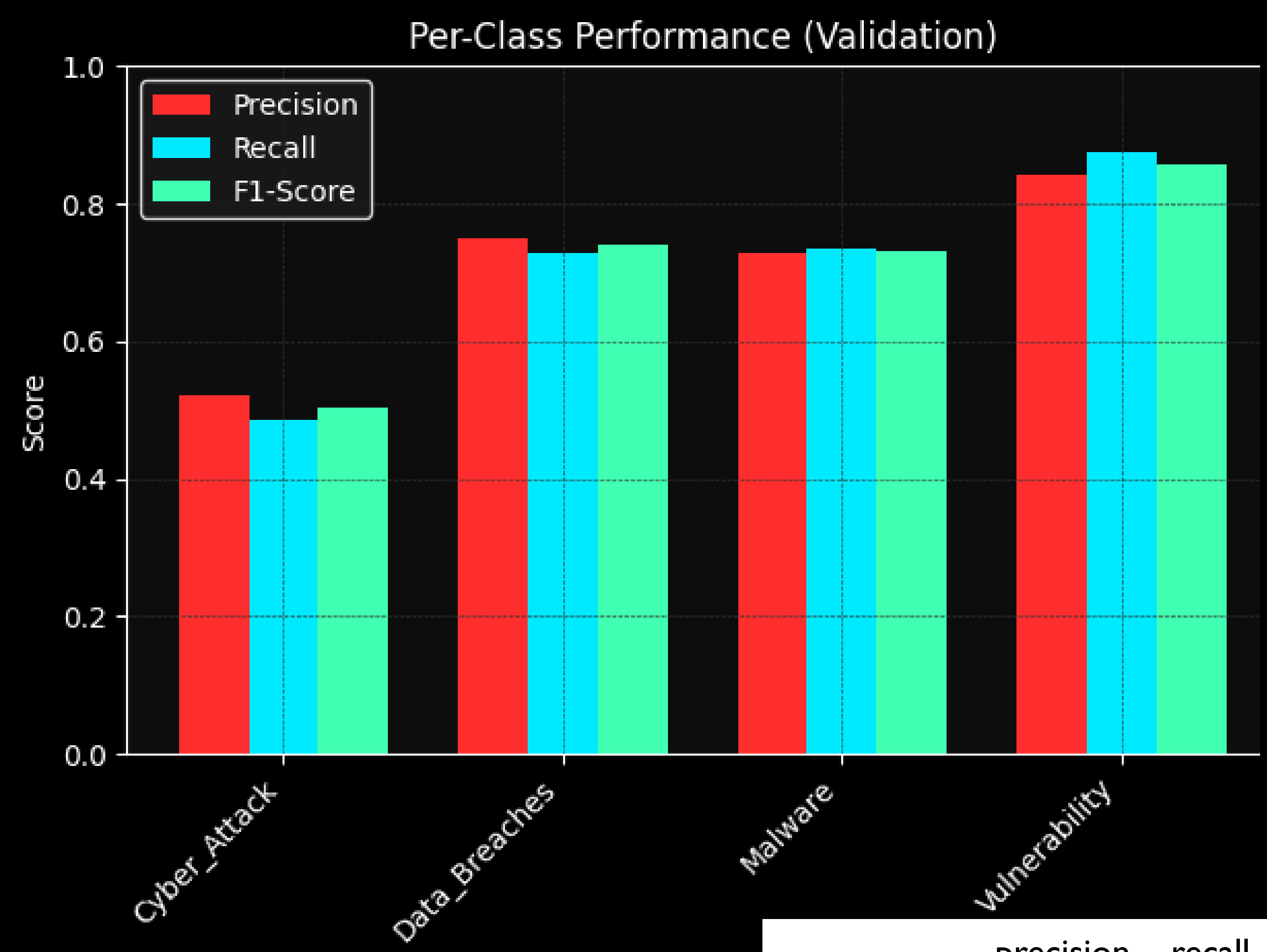
- Focal Loss – reduces the influence of majority class by downweighing confident predictions, helping stabilize training underrepresented classes for imbalanced datasets.

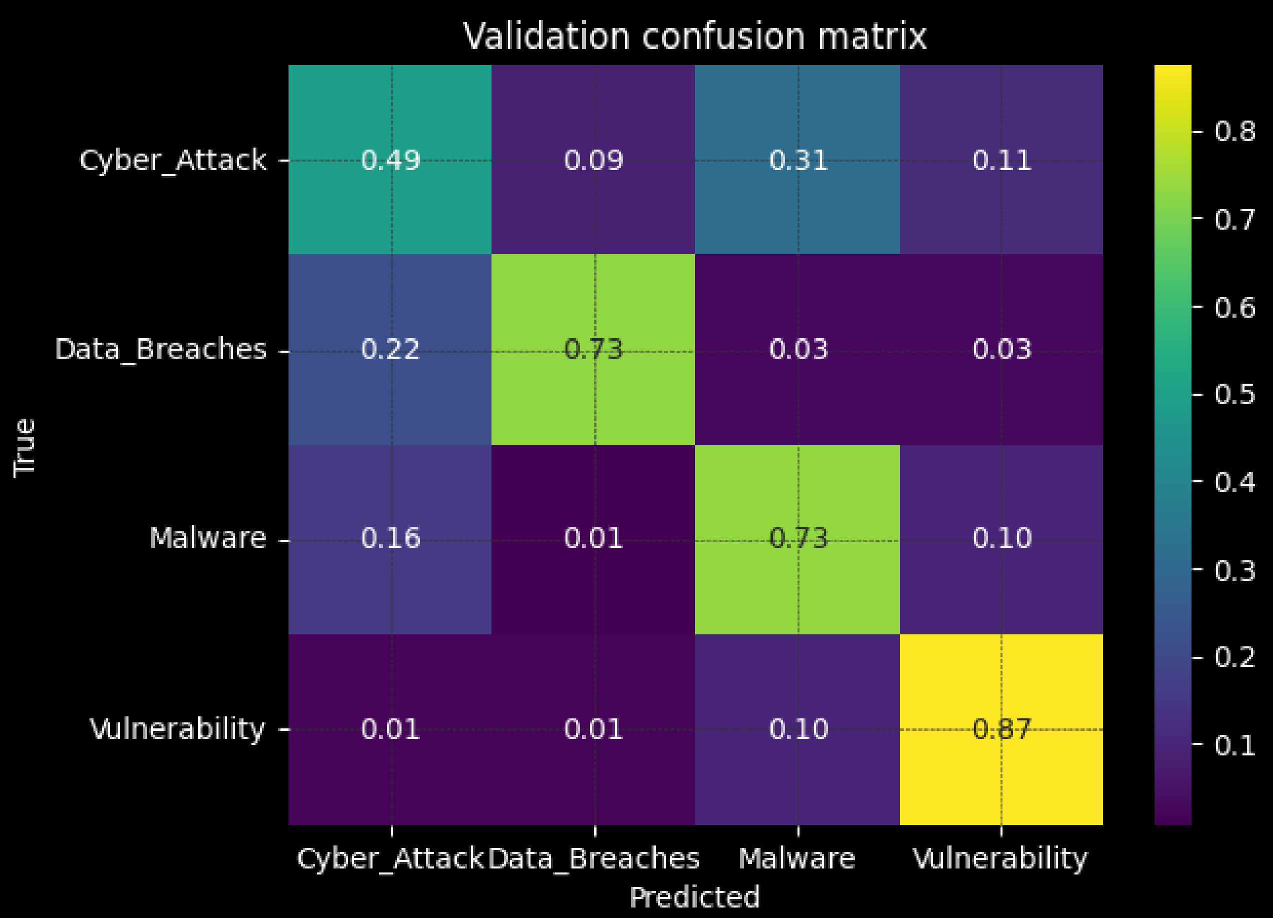
- Class weights – adds and adjusts weights so that errors on rare classes are penalized more than errors on common classes

Together the stabilize the training and improve macro-F1 performance.


Validation F1 per Epoch
F1 Score
0.67
0.68
0.69
0.70
0.71
0.72
0.73
Epoch
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00


Training loss over steps
Training Loss
0.6
0.7
0.8
0.9
1.0
1.1
1.2
1.3
Step
100
200
300
400
500





Validation

	precision	recall	f1-score	support
0	0.5231	0.4857	0.5037	70
1	0.7500	0.7297	0.7397	37
2	0.7293	0.7348	0.7321	132
3	0.8429	0.8741	0.8582	135
accuracy			0.7380	374
macro avg	0.7113	0.7061	0.7084	374
weighted avg	0.7337	0.7380	0.7356	374

Hyperparameter Tunning

The most important ones like learning rate or weight decay are optimized by using Optuna to run hyperparameter search while wrapping and controlling the training through the Hugging Face Trainer API

The optimal set after tuning is as follows:

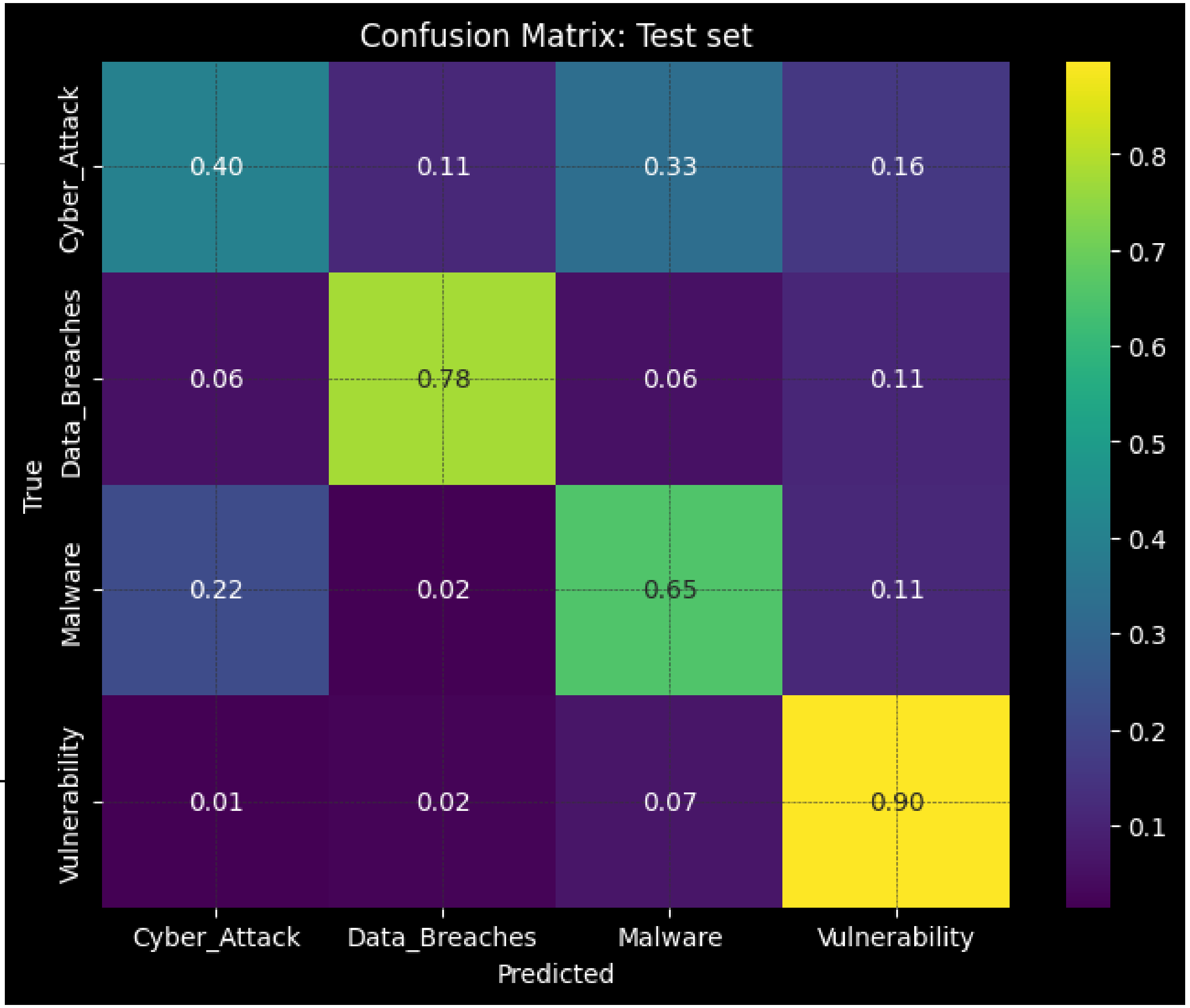
- Learning Rate: 2e-05
- Number of Trained Epochs: 3
- Warmup Ratio: 0.0
- Weight Decay: 0.01

Evaluating on the Test set

Classification Report

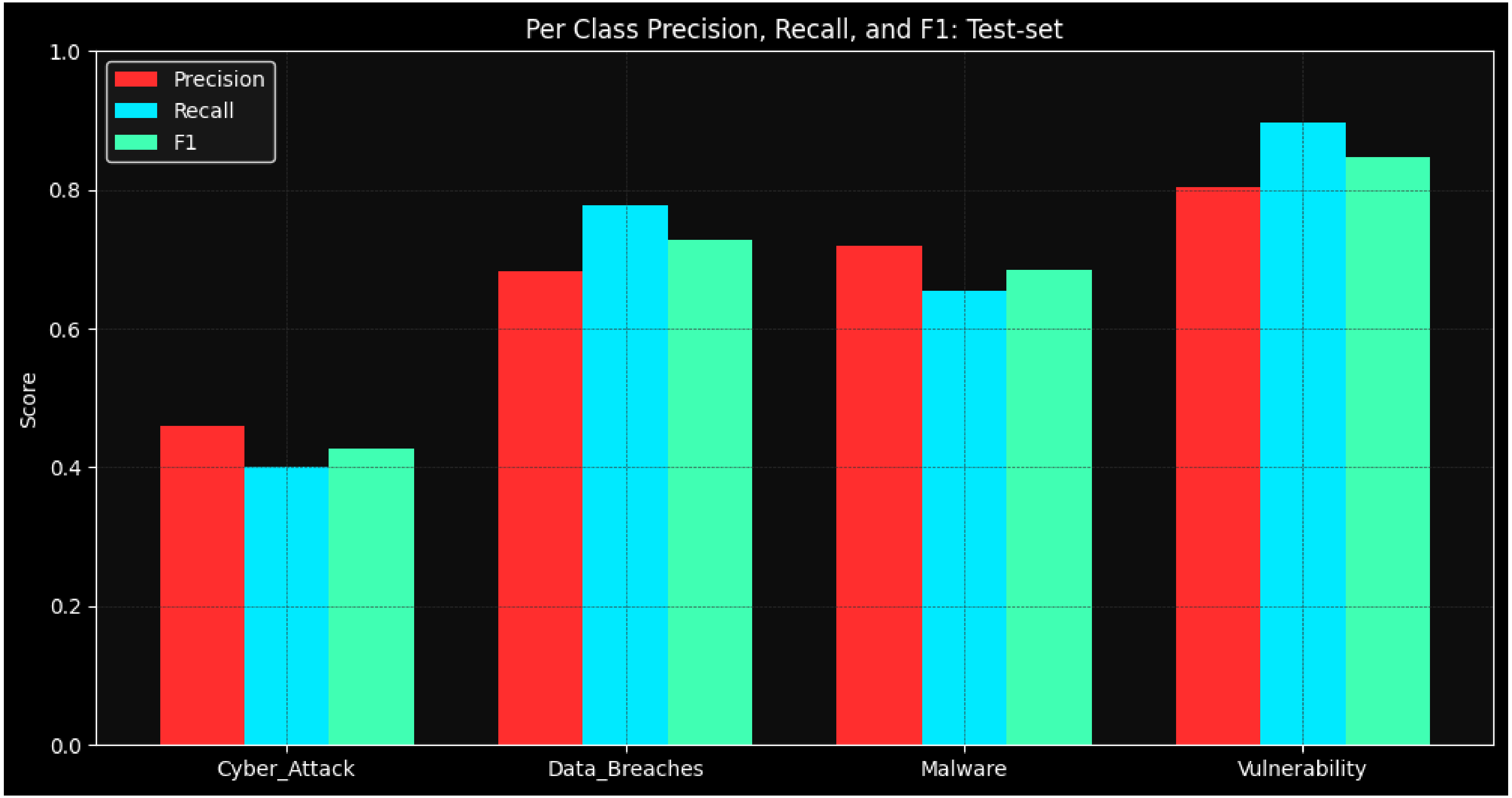
	precision	recall	f1-score	support
CA : 0	0.4590	0.4000	0.4275	70
DB : 1	0.6829	0.7778	0.7273	36
M : 2	0.7190	0.6541	0.6850	133
V : 3	0.8026	0.8971	0.8472	136
accuracy			0.7067	375
macro avg	0.6659	0.6822	0.6718	375
weighted avg	0.6973	0.7067	0.6998	375

- Class weights and Focal loss improved Data Breaches



True \ Predicted	Cyber_Attack	Data_Breaches	Malware	Vulnerability
Cyber_Attack	0.40	0.11	0.33	0.16
Data_Breaches	0.06	0.78	0.06	0.11
Malware	0.22	0.02	0.65	0.11
Vulnerability	0.01	0.02	0.07	0.90

Evaluating on the Test set



ML and DL Comparison

Ultimately a deep learning model should be able to outperform a classical ML model for text classification, but we don't see that in this comparison for 2 main reasons:

- Neural Networks work best on vast amounts of data (We only have 3742 records in our dataset)
- Large amount of semantic overlap between Malware and Cyber Attacks

With these 2 factors in mind, along with specific lexical features in ML and training time advantages, our ML model outperforms its DL counterpart for this dataset.

Practical Implications

This project is a Text-classification for cyber security corpora which can be utilized in the following ways within an organization:

- Automating classification of different incoming cybersecurity news usually done by analysts
  - Auto-sort large volumes of corpora
  - Saving labor and thus reducing costs

- Fast threat detection for increased response time in cyber security firms
  - Responding to threats/risks before they escalate

Questions?