TEXT-BASED CYBER THREAT DETECTION

MACHINE LEARNING FOR MALICIOUS TEXT CLASSIFICATION

AUTHORS: KELVIN KIPKORIR, LUCY MUTUA, CHARLES MUTEMBEI, SHARON AOKO, VICTOR MUSYOKI

PROBLEM STATEMENT

Rising text-based threats (phishing, malware-laced reports).

Manual detection is inefficient; rule-based systems fail to adapt.

Goal: Automate classification of text as *malicious* or *benign*.
Visual:

OBJECTIVES

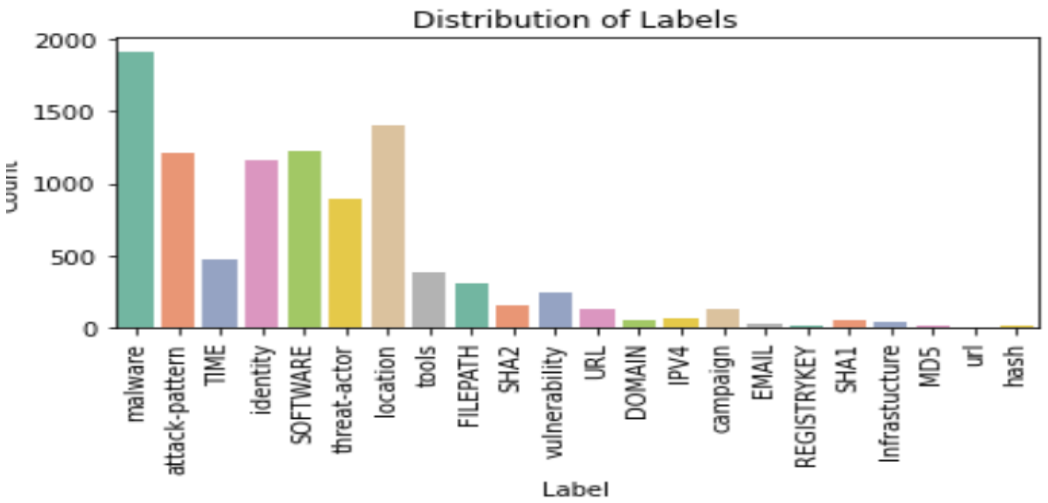
- **Primary Goal**:
  - Develop a machine learning model to classify cybersecurity text as *malicious* or *benign*.
- **Technical Approach**:
  - Leverage NLP techniques (tokenization, lemmatization) for text preprocessing.
  - Train and evaluate neural networks (NN, LSTM, BiLSTM) for threat detection.
- **Outcome Metrics**:
  - Achieve high accuracy (>90%) with balanced precision/recall.
  - Enable scalable, automated analysis of unstructured threat data.

Why these objectives will work:

- **They have a clear scope**: Focuses on the *what* (classification), *how* (NLP + ML), and *success criteria* (metrics).
- **The solution is audience-friendly**: Avoids jargon; aligns with both technical and non-technical stakeholders.
- **Provision of visualization**: Icons/flowcharts make the objectives memorable.

DATA UNDERSTANDING

- Source: Kaggle (19,000+ entries).
- Features: Raw text + entities (malware, tools, locations).
- Labels: malicious (9,938 samples) vs. benign (10,002).

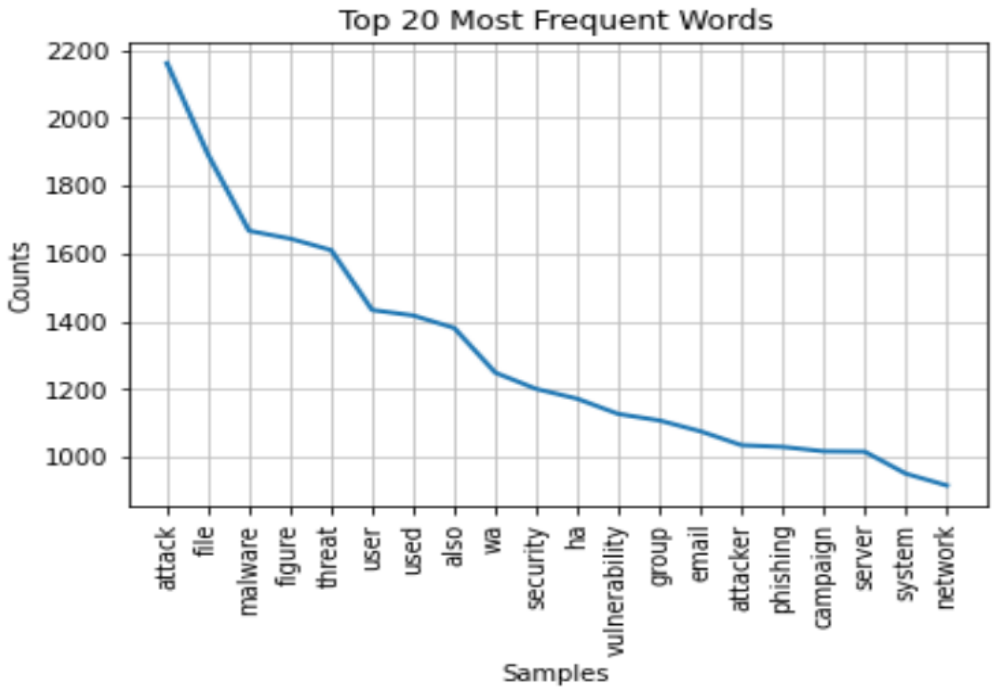


DATA PREPROCESSING

Cleaning & Preparing Text Data

Steps:

- Lowercasing, punctuation removal.
- Tokenization + lemmatization.
- Stopword removal.
- Padding sequences (max length = 250 tokens).



MODEL ARCHITECTURES

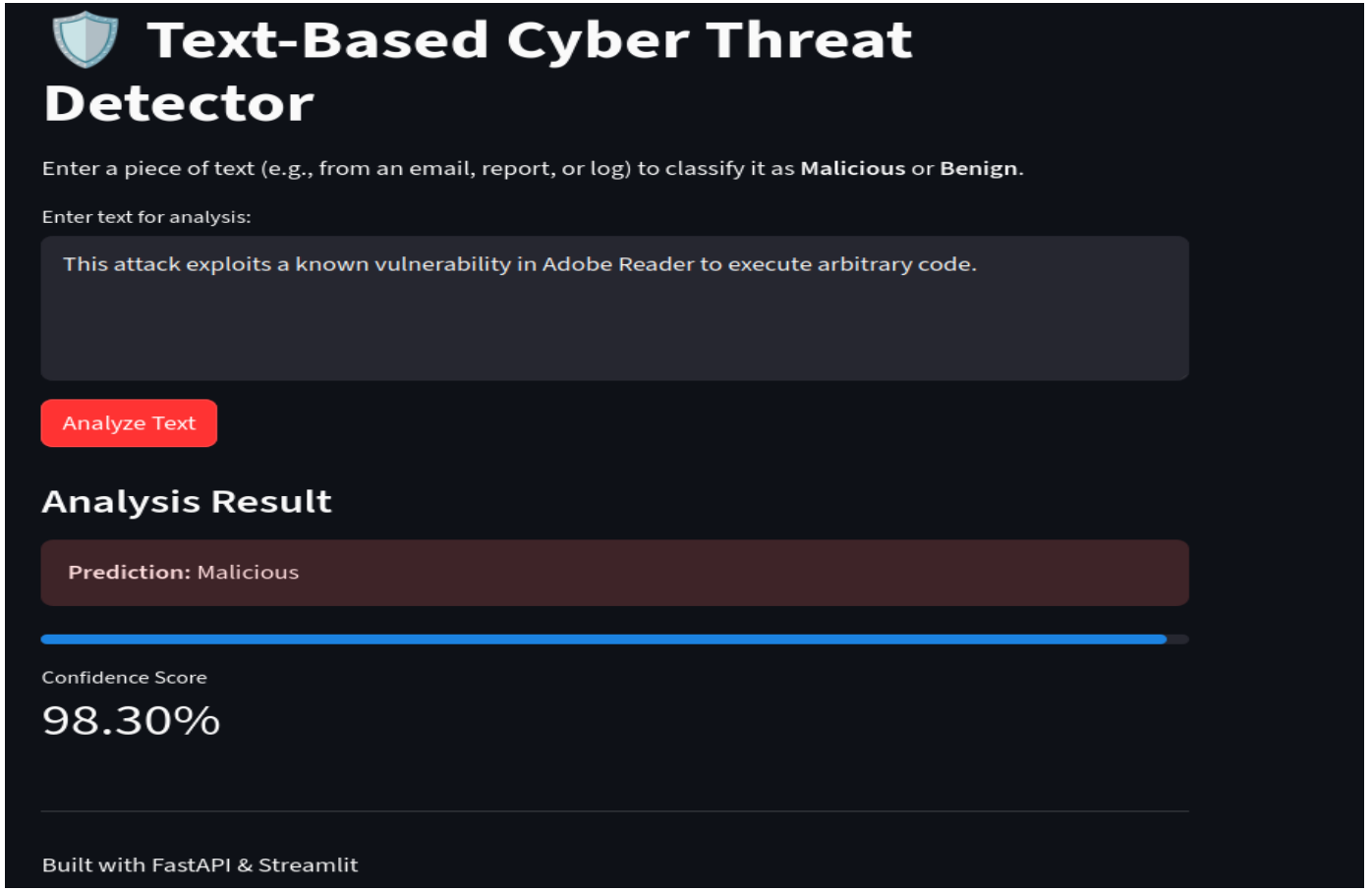
1. **Baseline NN**: Embedding → Flatten → Dense (95.3% value accuracy).

2. **Tuned NN**: Added dropout + extra dense layers (95.2% value accuracy).

3. **LSTM**: Poor performance (50% accuracy).

4. **Bidirectional LSTM**: Best results (95.3% value accuracy).

REAL-WORLD TEST



CHALLENGES & LESSONS

Challenges:

- LSTM’s failure (possible causes: insufficient tuning/data).
- False negatives in predictions.

Lessons:

- Bidirectional models capture context better.
- Dropout prevents overfitting.

FUTURE WORK

Experiment with Transformer Models

- We should aim to test with transformer models such as BERT or DistilBERT for improved contextual understanding.
- We should fine-tune on cybersecurity-specific text example threat reports, phishing emails to enhance accuracy for future.

Enhancement of Dataset Quality

- We should address false negatives by adding adversarial examples example obfuscated malware descriptions.
- We should include diverse threat types such as zero-day exploits, social engineering tactics, multilingual attacks.
- We should balance class distribution to reflect real-world scenarios for instance higher benign-to-malicious ratios.

Deploy Bidirectional LSTM (BiLSTM) with Feedback

- We should consider integrating the model into a user-friendly dashboard for security analysts.
- We should also implement a feedback loop to collect analyst corrections and iteratively improve the model.

Additional Improvements

- We should consider confidence thresholds that is flag low-confidence predictions for human review.
- We should consider real-time monitoring to adapt to evolving threats via continuous learning.

CONCLUSION AND RECOMMENDATIONS

Conclusion

Success Achieved:

- We managed to developed a prototype with 95.3% accurate Bidirectional LSTM model for text-based threat detection.
- We were able to demonstrated that NLP and deep learning can automate classification of malicious/benign text effectively.

Challenges Addressed:

- We overcame dataset limitations such as missing labels, class balance through careful preprocessing.
- We validated that sequential models (BiLSTM) outperform traditional NNs for contextual analysis.

Real-World Impact:

- We discovered there is potential to reduce analysts' workload by filtering benign content.
- We found out there is scalable solution for processing unstructured threat data such as emails, reports, logs.

Recommendations

Data Enhancements:

Collaborate with threat intelligence teams to expand dataset diversity for example dark web text, non-English threats. Implement continuous data labelling to keep the model updated.

Deployment Strategy:

Integrate the BiLSTM model into a SOC (Security Operations Centre) workflow as a first-pass filter. Design a feedback UI for analysts to correct misclassifications (active learning).

Monitoring & Maintenance:

Track model drift with real-time performance metrics. Schedule quarterly retraining with new threat data.

THANK YOU

GROUP 10

HTTPS://GITHUB.COM/KKIPKORIR/CYBER-THREAT-INTELLIGENCE