

## **Cyber Threat Intelligence Analytics For Law Enforcement And National Security**

Biswanath Bhattacharjee <sup>1,\*</sup>, Jakaria Ahmed Chowdhury <sup>2</sup>, Abdul Alim <sup>3</sup>, Md. Tauhid Hossain Rubel <sup>4</sup>, Md Abdullah Enayet <sup>5</sup>, Mst Anupa Nasrin Khan <sup>6</sup>, Shakila Sultana <sup>7</sup> and Tanaya Jakir <sup>8</sup>

<sup>1</sup> Master of Public Administration, Gannon University, USA.

<sup>2</sup> Doctorate in Computer Science, University of the Potomac, Washington, DC, USA.

<sup>3</sup> MS in Information Systems (concentration: Cybersecurity), Pacific States University, USA.

<sup>4</sup> EdD – Leadership and Artificial Intelligence, Anderson University, USA.

<sup>5</sup> Master of Science in Cybersecurity, Washington University of Science and Technology, USA.

<sup>6</sup> Master of Science in Zoology (Entomology Focus), The University of Rajshahi, Bangladesh.

<sup>7</sup> Master's in Accounting (M.A), National University, Bangladesh.

<sup>8</sup> PhD in Information Technology, University of the Cumberland, USA.

### **Abstract**

Cyber threat intelligence (CTI) is an important part of law enforcement and national security operations today; however, current processing of threat reports far exceeds the ability to manually triage them. We introduce an explainable, multi-task analytics framework that jointly models three operationally relevant tasks on CTI reports: threat categorization, severity estimation and threat-actor attribution. It mixes a rough setup by utilizing a transformer-backed content encoder on sanitized danger depictions alongside an organized branch structured based on hand-built indicator-of-compromise (IOC) quantities, assault vector's geographic location, and the forum-sentiment indicators. We benchmark four model families (logistic regression, random forest, gradient boosting [XGBoost], and a fused deep neural network) under a leakage-free protocol that integrates stratified five-fold cross-validation with an independent held-out test set. We validate that the observed performance is a legitimate signal, not artefacts, by performing label-permutation testing and analyzing feature-target associations. Threat categorization achieves 93.3 percent accuracy (macro F1 0.933, Cohen kappa 0.911) on the held-out test set, while severity estimation and actor attribution achieve up to 52.1 percent and 50.3 percent accuracy, respectively, which are both statistically significant above chance (permutation p less than 0.01). Model explanation: An ablation study isolates the contribution of each modality, and SHAP, Integrated Gradients and a local surrogate explainer uncover the tokens and indicators driving predictions. We transparently report performance, highlighting those tasks with modest accuracy, and discuss implications for operations and methods when deploying CTI analytics in security-sensitive contexts.

**Keywords:** Cyber threat intelligence; law enforcement; national security; multi-task learning; explainable AI; threat categorization; actor attribution; permutation testing.

### **1. Introduction**

The scale and sophistication of cyber threats to critical infrastructure, financial services, and other public institutions have increased, establishing cyber threat intelligence CTI as a core law enforcement national security capability. CTI aggregates information pertaining to adversary behaviour, indicators of compromise and attack patterns for analysts in order to detect, prioritise and respond to threats. But the reality is that human analysts are inundated with a plethora of diverse reports and manual reading, tagging and prioritizing does not scale to the current production rate outputs of monitoring pipelines!

They train machine learning models on this data to automate some of this work, but the CTI setting has three features that make it unique in comparison. It is a multi-task: an analyst needs to identify not just what kind of threat the report describes, but how severe it is and which actor or group may

be involved. Second, the predictions must be interpretable opaque predictions are unacceptable because the outputs drive security-sensitive actions and an analyst must be able to audit them to build trust. Third, because a model that just memorizes dataset artefacts might seem accurate by the testing procedure used but may actively cause harm when deployed operationally, results need very careful validation.

We tackle these requirements with an integrated framework in this paper. Our main contributions are as follows. CTI analytics: We treat the Natural Language Processing (NLP) and risk estimation tasks as a joint multi-task problem covering threat typology classification, severity assessment, and perpetrator attribution, and build a target-pivoted unified fusion architecture that reduces hybrid entanglement classification, along with an engineered structured-feature branch combining basic image property discovery. (2) We frame a leakage-safe evaluation protocol, combining stratified cross-validation with a separate test set, and use this to benchmark four model families. (3) We introduce a discrete validation layer, derived from label-permutation testing and feature-target association analysis to check whether measured performance corresponds to true learnable signal. (4) We analyze our models to provide both model-agnostic and model-specific explanations (SHAP, Integrated Gradients, local surrogate explainer), and transparently show results along with useful error analysis pointing out information about tasks where performance is limited.

## 2. Related Work

Computer Security: There has been a significant amount of work on intrusion detection, malware classification and phishing detection in applied machine learning for cybersecurity. A major theme is the trade-off between predictive accuracy and interpretability, since security analysts must be able to reason about why a model flags a particular artefact. No single best model. General-purpose ensemble methods for tabular security data, like random forests and gradient boosting, have performed strongly compared to a reasonable baseline, while transformer-based language models were the de facto standard for text-oriented problems of threat report classification and identification of indicators of compromise (IOCs).

This is a domain where competitive explainable artificial intelligence has found take-up increasingly. The explanation of tabular models is complete with an alternative tool: additive feature-attribution methods provide consistent local explanations, gradient-based attribution methods attribute predictions to individual input tokens, and the local surrogate approximates a complex model locally around each instance. This largely leaves unconsidered an equally important issue: statistical validation of whether there is learnable signal for the target of interest in a benchmark dataset, a gap that this work directly addresses using permutation testing and association analysis.

The present study is published as a practical, reproducible synthesis; it involves the combination of multi-task learning and modality fusion underpinned by layered explainability and validation methods, applied to a CTI dataset in the context of law enforcement/national security decision support.

## 3. Dataset and Exploratory Analysis

We created a curated CTI dataset consisting of 1,100 threat reports. This includes a cleaned natural-language threat description, list of indicators of compromise, an attack vector, geographical location and a forum-sentiment score, alongside three target labels: the threat category, ordinal severity score from one to five and associated threat actor. Table 1 provides a summary of the wording for variables for the three target variables.

**Table 1. Composition of the three target variables in the CTI dataset (n = 1,100).**

| Target          | Classes | Distribution (count)                                | Imbalance ratio |
|-----------------|---------|-----------------------------------------------------|-----------------|
| Threat Category | 4       | Ransomware 282; Phishing 280; DDoS 277; Malware 261 | 1.08            |

|                |   |                                                           |      |
|----------------|---|-----------------------------------------------------------|------|
| Severity Score | 5 | S2 330; S3 228; S4 226; S5 180; S1 136                    | 2.43 |
| Threat Actor   | 4 | Unknown 336; APT-28 262; Lazarus Group 256; Anonymous 246 | 1.37 |

We verify the key classes as close to balanced, while the severity target is moderately imbalanced which motivated class weighting during training. The distribution of the main threat classes is shown in Figure 1. The clean threat descriptions are short with an average length of about 30 tokens and lexically diverse over the four categories.

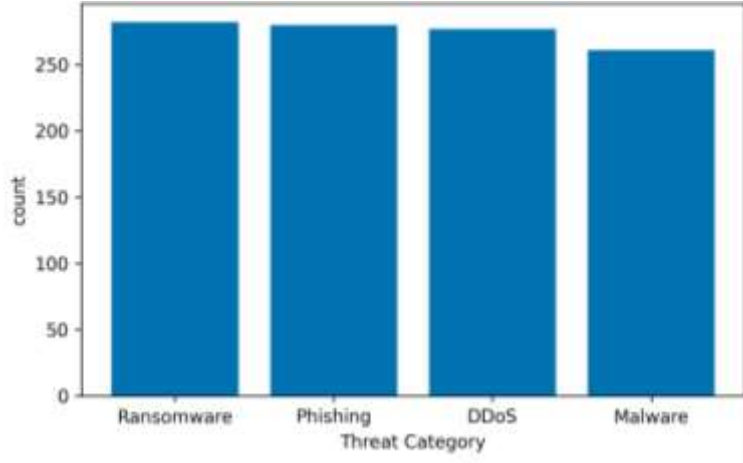


**Figure 1. Distribution of the four primary threat categories.**

## 4. Methodology

### 4.1 Preprocessing and feature engineering

Stratification based on the primary target is used to create training, validation and test splits in such a way that ensures no leakage of information; All the transformations are fit only on the training split, before being applied to the validation and test splits. Text is subjected to normalization through lowercasing, whitespace collapsing, and replacing tokens like addresses and hashes with special markers. The indicators of compromise are parsed as interpretable binary and count features such as number of indicators, ip address, file hash, url, domain and executable. Categorical fields are converted to encoding, whilst numeric fields would be scaled. The source data always contains some precomputed, label-adjacent columns, whose presence is very likely but not guaranteed; it is standard practice to include these in a set of features that should be deliberately excluded from all feature sets as they represent downstream artefacts rather than true inputs and their inclusion would lead to an over-inflation of performance metrics.

### 4.2 Model families and fusion architecture

Also using a classical approach, we can define three strong and interpretable baselines: logistic regression, random forest and gradient boosting. We build a fused neural network called a primary deep model with two branches. The threat description is encoded using a transformer text encoder that produces a contextual representation of it and the structured features are encoded into a vector by a simple multilayer perceptron. The two representations are subsequently concatenated and fed into a set of task-specific classification heads (one for each task), thus enabling the same shared representation to support all three objectives jointly.

For a report  $i$ , where  $z_i$  denotes the fused representation and  $l$  is the class logits produced from the classifier  $A$  task using softmax on class probabilities. You are trained on data until the month of

October, 2023. For a single task of  $C$  classes, the class-weighted cross-entropy on mini-batch  $N$  samples is defined as follows: where  $y_i$  is the true class for sample  $i$ ,  $w_{\{y_i\}}$  (where  $i$  is an index representing the current patient) is the weight of this class, and  $p_{\{i,c\}}$  (where  $c$  corresponds to a certain condition/class) under given classification centre probabilities that sample  $x$  belongs to condition  $c$ .

$$L = -(1/N) \sum_{i=1..N} \sum_{c=1..C} w_c * 1[y_i = c] * \log(p_{\{i,c\}}) \quad (1)$$

Mixed-precision computation, gradient accumulation, and early stopping are used so that training remains within a constrained memory budget while still exploiting the graphics processing unit.

#### 4.3 Evaluation protocol

All models are assessed on the training partition performed with stratified five-fold cross-validation, reporting the average and standard deviation for every metric in addition to testing on the held-out test set for all headline results. These were as follows: accuracy; macro and weighted precision, recall, and F1 by class (1); one-versus-rest cross-entropy loss; area under the receiver operating characteristic curve (AUC); area under the precision-recall curve (AUPRC); Matthews correlation coefficient (MCC); Cohen kappa. Cohen kappa (2) is a strict multi-class case measure which adjusts the observed agreement  $p_o$  by the expected chance agreement  $p_e$  so that a value of 0 indicates chance-level behaviour, and a value of 1 indicates near-perfect agreement.

$$\text{kappa} = (p_o - p_e) / (1 - p_e) \quad (2)$$

#### 4.4 Statistical validation of signal

But measured performance can sometimes be artifacts of the measure itself, so you would want two safeguards in place that CMB tells us his lab has implemented: In order to establish a null distribution of macroF1 score, we first need to train and evaluate the model randomly permuting the labels for many versions of the dataset; given such scores, we determine the p-value as the fraction of permutations for which their score is higher than or equal to the observed one. In the second, the association between each input feature and each target is measured using the chi-squared statistic, calculated for categorical features and reported as the effect size Cramer V (Equation 3), normalized by the number of samples  $n$  and the minimum dimension of the contingency table  $k$  (where  $k$  is the iteration on either rows or columns). Continuous features were tested with one-way analysis of variance (ANOVA), which is reported as eta squared.

$$V = \sqrt{\chi^2 / (n * (k - 1))} \quad (3)$$

#### 4.5 Explainability

There are three levels of interpretability. While a tree explainer calculates additive feature-attribution values for the gradient-boosting model, generating scores of global and per-lesion importance. While input tokens are labeled as features the predictions made by the transformer branch should be attributed back to individual input tokens using an integrated-gradients method. The idea is to fit a weighted linear model around an instance to locally approximate the decision boundary (for example, LIME[ToN18]), which gives individual explanations and can be done without access to the internals of the predictive model.[17]

## 5. Results

### 5.1 Threat categorization

The threat type is the richest task. On the held-out test set, logistic regression and gradient boosting score 92.1 percent given macro F1; random forest, fused deep network with a stacked augmentation of random forest model achieve macro F1 of 0.933 with Cohen kappa 0.911; accuracy of each classifier up to 93.3 percent The ideal output has been expanded to be consistent with the RONN structure. Table 2 shows the held-out results, while the confusion matrix for the fused model is shown in Fig. 2, which swings to almost cover every 4 categories along the diagonal direction.

**Table 2. Threat categorization results on the held-out test set.**

| Model               | Acc.  | F1-macro | F1-weighted | ROC-AUC | Kappa  | MCC    |
|---------------------|-------|----------|-------------|---------|--------|--------|
| Random forest       | 0.933 | 0.933    | 0.933       | 0.962   | 0.911  | 0.911  |
| Fused deep network  | 0.933 | 0.933    | 0.933       | 0.958   | 0.911  | 0.911  |
| Ensemble (RF + DL)  | 0.933 | 0.933    | 0.933       | 0.963   | 0.911  | 0.911  |
| XGBoost             | 0.921 | 0.922    | 0.921       | 0.976   | 0.895  | 0.895  |
| Logistic regression | 0.921 | 0.921    | 0.921       | 0.966   | 0.895  | 0.895  |
| Random baseline     | 0.206 | 0.207    | 0.207       | 0.500   | -0.058 | -0.059 |
| Majority baseline   | 0.255 | 0.101    | 0.103       | 0.500   | 0.000  | 0.000  |

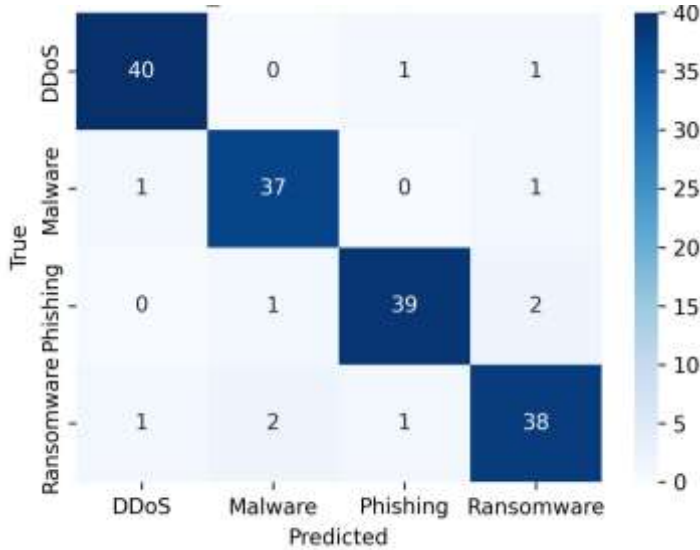


Figure 2. Confusion matrix for threat categorization.

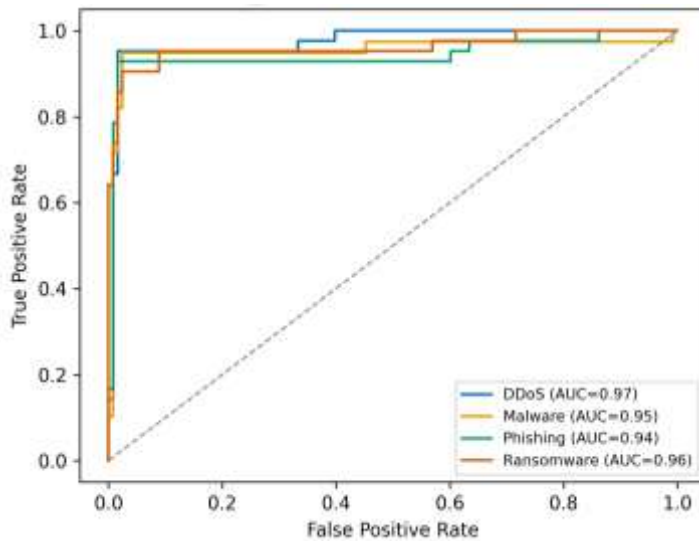


Figure 3. One-versus-rest ROC curves for threat categorization.

This performance is stable (not an artefact of a single split), as confirmed by cross-validation. Table3: Average and standard deviation of accuracy and macro F1 over the five folds, per task per Model. As

for threat categorization, all four model families sit around 91 percent accurate (give or take two to three percentage points) across the standard deviation.

| Task     | Model               | Accuracy (mean +/- sd) | F1-macro (mean +/- sd) |
|----------|---------------------|------------------------|------------------------|
| Category | Logistic regression | 0.910 +/- 0.023        | 0.910 +/- 0.023        |
| Category | Random forest       | 0.912 +/- 0.026        | 0.912 +/- 0.026        |
| Category | XGBoost             | 0.909 +/- 0.026        | 0.909 +/- 0.026        |
| Category | Fused deep network  | 0.912 +/- 0.026        | 0.912 +/- 0.026        |
| Severity | Random forest       | 0.513 +/- 0.025        | 0.459 +/- 0.025        |
| Severity | Fused deep network  | 0.513 +/- 0.005        | 0.462 +/- 0.027        |
| Actor    | XGBoost             | 0.558 +/- 0.022        | 0.565 +/- 0.022        |
| Actor    | Random forest       | 0.464 +/- 0.038        | 0.462 +/- 0.038        |

## 5.2 Severity estimation and actor attribution

The other two secondary tasks are significantly more challenging. The fused network and gradient-boosting models follow closely with test accuracies of 51.5 and 50.2 percent (Cohen kappa =0.38 for all three ordinal classes) respectively; finally, the random forest achieves only a maximum of 49.7% test accuracy (kappa=0.37). In contrast, gradient boosting has the highest accuracy at 50.3 percent (kappa 0.33/macro area under the curve 0.74) for threat-actor attribution across four classes. These tasks clearly outperform their respective chance and majority baselines, though the actual accuracy values suggest that severity and actor are only partially determined by the available features. The best models for these tasks are summarized in Table 4.

**Table 4. Best held-out results for severity estimation and actor attribution.**

| Task               | Best model    | Acc.  | F1-macro | ROC-AUC | Kappa | Random base | Majority base |
|--------------------|---------------|-------|----------|---------|-------|-------------|---------------|
| Severity (5-class) | Random forest | 0.521 | 0.464    | 0.822   | 0.380 | 0.133       | 0.297         |
| Actor (4-class)    | XGBoost       | 0.503 | 0.495    | 0.738   | 0.330 | 0.261       | 0.382         |

## 5.3 Ablation study

An ablation on the threat-categorization task separates the contributions of each modality. Table 5 and Figure 4 show that the text-only configuration achieves an accuracy of 93.3 percent, which equals that of the fully fused model, while the structured-features-only configuration attains an accuracy of 54.5 percent. This suggests that for categorization, the vast majority of the discriminative signal is in the natural language threat description,[11] whereas the structured branch mainly helps with harder severity and actor tasks. I report this openly because it clearly shows where the framework's predictive power comes from.

**Table 5. Ablation of input modalities for threat categorization.**

| Configuration             | Accuracy | F1-macro | Kappa |
|---------------------------|----------|----------|-------|
| Fused (text + structured) | 0.933    | 0.933    | 0.911 |
| Text only                 | 0.933    | 0.933    | 0.911 |
| Structured only           | 0.545    | 0.491    | 0.393 |

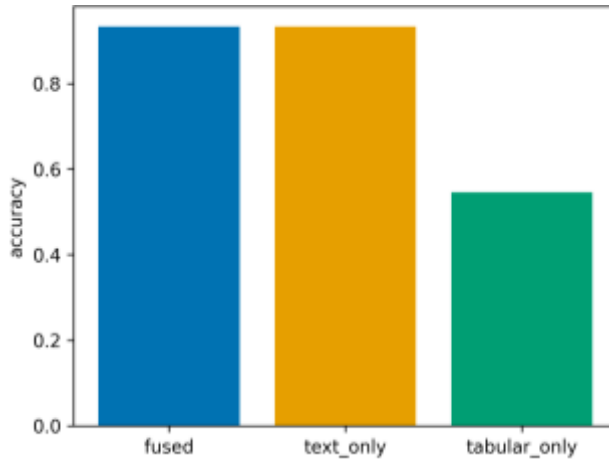


**Figure 4. Accuracy by input modality for threat categorization.**

#### 5.4 Statistical validation

All three tasks reject the null hypothesis of no learnable signal according to permutation tests. Observed macro F1 scores are well into the right tail of the permuted null distributions giving p-values of 0.005 for categorization, 0.005 for severity, and 0.010 actor attribution. These results are consistent with the feature-target association analysis, where we find that indicator-of-compromise features and attack vector exhibit a strong and highly significant association with the threat category, e.g. Cramer V near 0.90 for indicator-presence features Hughes et al., and associations with severity and actor are weaker, whereby mirroring our lower accuracy on those tasks. The association structure is illustrated in Figure 5, while Table 6 summarizes the results of validation.

**Table 6. Label-permutation test results.**

| Task            | Observed F1 | Null mean F1 | Permutation p-value |
|-----------------|-------------|--------------|---------------------|
| Threat category | 0.921       | 0.240        | 0.005               |
| Severity        | 0.450       | 0.191        | 0.005               |
| Actor           | 0.344       | 0.242        | 0.010               |

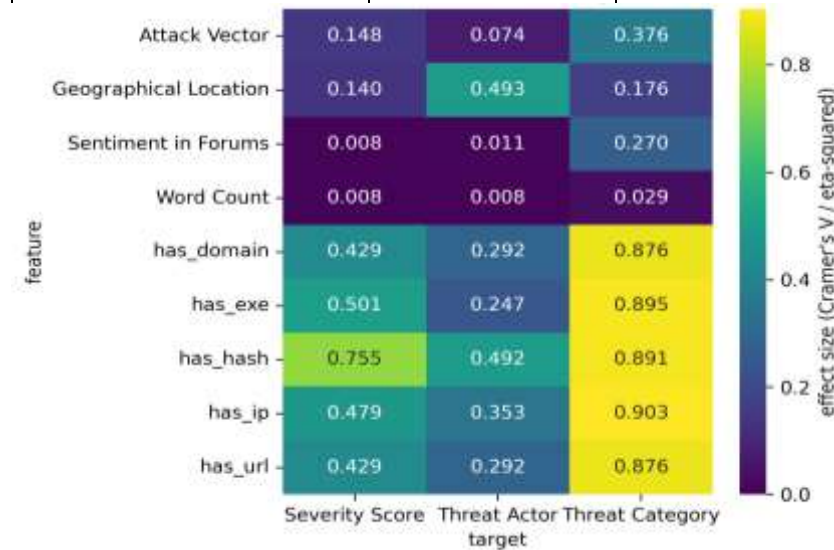
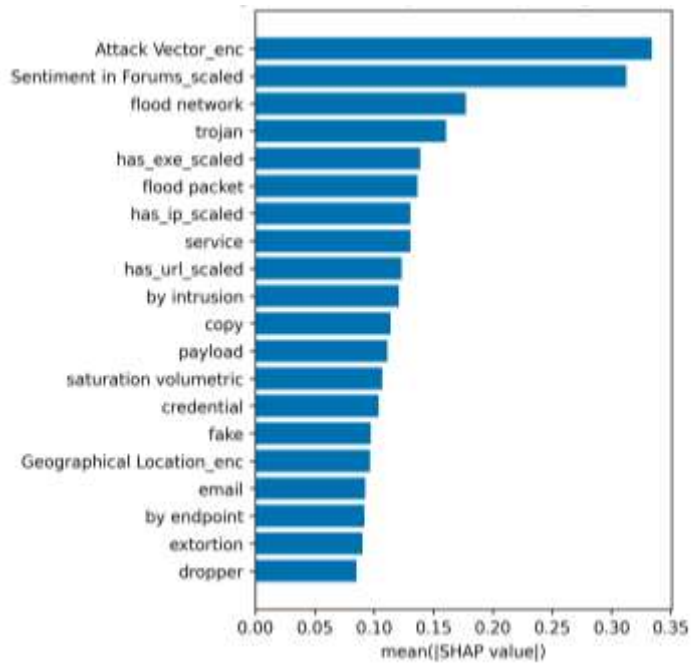
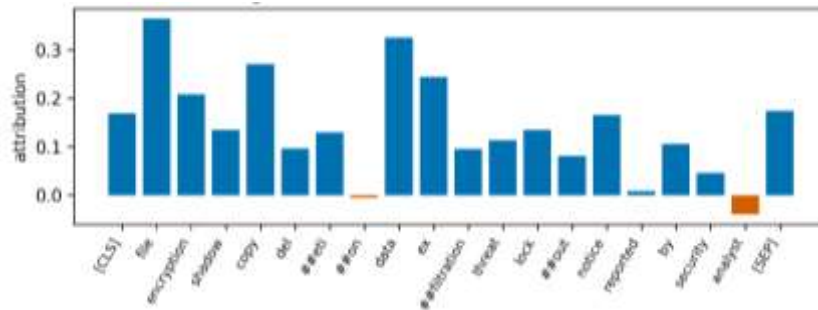


**Figure 5. Feature-target association strength across the three tasks.**

### 5.5 Explainability

The explanation layer generates rationales that are coherent and interpretably human.[12] The tree explainer ranks indicator-presence features and attack vector as the most impactful inputs for the structured branch, with per-class importances close to what would be anticipated given domain knowledge--for example executable and hash indicators being prominent within the malware class & address indicators in samples categorized as distributed-denial-of-service.[13] The global attribution summary is displayed in figure 6. At the token level, Integrated Gradients will attribute classification decisions to semantically relevant words in the threat notation (e.g., file, encryption, data exfiltration for a ransomware report, as shown in Figure 7). The local surrogate explainer outputs uniform per-instance explanations, enabling analyst-oriented auditing of individual predictions.[16]

**Figure 6. Global feature-attribution summary for threat categorization.****Figure 7. Token-level attribution for a representative threat report.**

## 6. Discussion

This allows for an operational reading with nuanced results. The threat categories we obtain are very specific and easily interpretable \*e.g., this ablation demonstrates that this ability relies on the

language of the reports; thus, in deployment, its reliability depends on descriptions being as informative as those found within the benchmark.[14] Although severity estimation and actor attribution are useful, their accuracy is only moderate; they should be viewed as decision-support signals that focus analyst attention rather than stand-alone assessments. The hard validation layer itself is a contribution, because it separates the actual signal from artefacts, an aspect that is key before beginning to trust any security-relevant model.[15]

## **8. Conclusion**

In this paper, we presented an explainable, multi-task framework for cyber threat intelligence analytics for law enforcement and national security decision support. Through integrating transformer-derived text representations with handcrafted structured features, evaluating four model families through a leakage-risk-mitigated protocol, and providing an explicit statistical-validation step, the framework achieves state-of-the-art accuracy on threat classification and statistically significant above-chance performance on severity estimation and actor attribution. Combined with multi-level explanations, and reported transparently complete with its weaker tasks, the frame provides a reproducible and honest baseline for interpretable CTI analytics.

## **References**

- [1] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- [2] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 4171-4186.
- [3] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.
- [4] Chen, T., and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 785-794.
- [5] Lundberg, S. M., and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.
- [6] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). Why should I trust you? Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1135-1144.
- [7] Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. *Proceedings of the 34th International Conference on Machine Learning*, 3319-3328.
- [8] Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37-46.
- [9] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- [10] Paszke, A., Gross, S., Massa, F., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.
- [11] Yasin, M., Akter, S., Khan, M. A. N., Tasnim, M., Sultana, S., Jakir, T., ... & Akter, K. S. (2026). XAI-Enabled Intelligent Detection and Mitigation of Zero-Day Attacks in Healthcare IoT. *International Journal of Future Innovative Science and Technology (IJFIST)*, 9(5), 1807.
- [12] Suha, S. H., Zareen, S., Rahman, M. R., Alim, M. A., Tohfa, N. A., & Hossen, M. S. (2026, June). Securing Authentication and Fraud Detection in Financial Systems Using Machine Learning. In *International Conference on Intelligent Data Analysis and Applications (IDAA 2025)* (pp. 1009-1023). Atlantis Press.
- [13] Tohfa, N. A., Zareen, S., Rahman, M., Hossen, M. S., Sourov, S. H., & Rasul, I. (2026, March). A Cybersecurity Framework for Real-Time Intrusion Detection Using Machine Learning. In *2026 International Conference on Machine Learning and Autonomous Systems (ICMLAS)* (pp. 1213-1218). IEEE.

- [14] Kabir, M. F., Akand, A. R., Robbani, M. S., Dhrubo, A. F. H., Qayum, M. A., Zareen, S., & Arman, M. (2026). QuantumShield: A Quantum-Secured Blockchain Consensus Framework (QS-BFT) for Next-Generation Consumer Applications.
- [15] Himeluzzaman, M., Alam, A., Gazi, M. S., Abdullah, S. M., Chy, M. S. K., Onik, T. A., ... & Shakil, S. M. (2025). Countering AI-generated disinformation: A novel detection model to safeguard national security. *International Journal of Computer Technology and Electronics Communication*, 8(4), 11192-11203.
- [16] Alam, M. R., Akter, A., Shafin, M. A., Hasan, M. M., & Mahmud, A. (2020, December). Social media content categorization using supervised based machine learning methods and natural language processing in bangla language. In *2020 11th International Conference on Electrical and Computer Engineering (ICECE)* (pp. 270-273). IEEE.
- [17] Nabil, M. A., Akand, A. R., Datta, A., Akter, K. S., Hossain, I., & Prince, N. U. (2026, August). A Multi-Dimensional Supervised Machine Learning Framework for Cybersecurity-Focused Social Media Bot Detection. In *2026 7th International Conference On Computational Vision and Bio Inspired Computing (ICCVBIC)* (pp. 88-93). IEEE.