Naif Arab University for Security Sciences
Arab Journal of Forensic Sciences and Forensic Medicine
المجلة العربية لعلوم الأدلة الجنائية والطب الشرعي
https://journals.nauss.edu.sa/index.php/AJFSFM

AI-Driven Forensics Framework for Reliable Detection of Ransomware

إطار عمل للتحقيق الجنائي المدعوم بالذكاء الاصطناعي للكشف الموثوق عن برمجيات الفدية

Mohammed Ahmed Alshalfi[1], Abdulrahman Mohammed Alshehri[1], Mohammed Mana Almunajam[1], Saud Thabet Almatrafi[1], and Qazi Emad Ul Haq[1,2*]

[1]Department of Cybersecurity and Digital Forensics, College of Forensics and Investigative Sciences, Naif Arab University for Security Sciences, Riyadh, Saudi Arabia

[2]Centre of Artificial Intelligence for Security (CAIS), Naif Arab University for Security Sciences, Riyadh, Saudi Arabia

Received 16 Feb. 2026; accepted 30 Mar. 2026; available online 10 Sep. 2026

Abstract

Ransomware is a significant cybersecurity risk that keeps increasing and threats like the WannaCry keep showing the need to be more intelligent and adaptive in detection methods. This paper presents DeepRansom, a powerful deep learning model that has been designed to detect WannaCry ransomware in complex malware environments with high precision and in a transparent manner. The architecture combines the feature selection of TabNet and the learning context of the FT-Transformer to create a dual-model architecture that can provide performance and interpretability. DeepRansom was trained and tested on a broad dataset of 21,752 samples of 26 malware families with 11 key ransomware variants. The proposed model reformulates WannaCry recognition as a binary classification task and delivers better results than the classical algorithms including Random Forest, SVM, and XGBoost among other deep learning baselines. The experiments have proven accuracy of 99.82%, precision of 97.89%, recall of 93.94% and AUC of 0.99% with strong generalization and low false positives. In addition, the SHAP-based analysis of interpretability shows that the predictions obtained by DeepRansom are based on the appropriate behavioral and structural predictors, which supports the idea of its implementation in the practice of network forensics and threat intelligence missions.

Keywords: forensic sciences, cybersecurity, deep learning, FT-Transformer, network forensics, WannaCry ransomware

المستخلص

تُشكل برمجيات الفدية (Ransomware) خطراً كبيراً على الأمن السيبراني في تزايد مستمر، وتستمر تهديدات مثل «واناكراي» (WannaCry) في إظهار الحاجة الملحة لأساليب كشف أكثر ذكاءً وتكيفاً. تقدم هذه الورقة نموذج DeepRan-som، وهو نموذج قوي يعتمد على التعلم العميق، صُمم للكشف عن برمجية الفدية «واناكراي» في بيئات البرمجيات الخبيثة المعقدة بدقة عالية ونهج شفاف.

تجمع بنية النموذج بين آلية اختيار الميزات في شبكة (TabNet) وسياق التعلم الخاص بـ(FT-Transformer) لإنشاء بنية نموذج مزدوجة توفر الأداء العالي والقابلية للتفسير في آن واحد. تم تدريب واختبار نموذج DeepRan-som على مجموعة بيانات واسعة تشمل 21,752 عينة لـ 26 عائلة من البرمجيات الخبيثة، مع 11 متغيراً رئيسياً لبرمجيات الفدية.

يعيد النموذج المقترح صياغة عملية التعرف على «واناكراي» كمهمة تصنيف ثنائي، ويقدم نتائج أفضل من الخوارزميات الكلاسيكية بما في ذلك (Random Forest) و(SVM) و(XGBoost)، بالإضافة إلى نماذج التعلم العميق المرجعية الأخرى. وقد أثبتت التجارب تحقيق دقة بلغت 99.82%، وضبط بنسبة 97.89%، واستدعاء بنسبة 93.44%، ومنطقة تحت المنحنى (AUC) بلغت 0.99%، مع قدرة قوية على التعميم وانخفاض في النتائج الإيجابية الخاطئة.

بالإضافة إلى ذلك، يُظهر تحليل القابلية للتفسير القائم على قيم (SHAP) أن التنبؤات التي حصل عليها DeepRansom تستند إلى مؤشرات سلوكية وهيكلية ملائمة؛ مما يدعم فكرة تنفيذه في ممارسات التحقيق الجنائي للشبكات ومهام استخبارات التهديدات.

الكلمات المفتاحية: علوم الأدلة الجنائية، الأمن السيبراني، التعلم العميق، FT-Transformer، التحقيق الجنائي للشبكات، برمجية الفدية «واناكراي»

* Corresponding Author: Qazi Emad Ul Haq
Email: qabdulrab@nauss.edu.sa

doi: 10.26735/QOQL7032

1. Introduction

Ransomware has become an urgent issue in cybersecurity within a very short period of time, impacting both individuals, businesses, and government infrastructure [1-3]. As one of its numerous variations, the WannaCry was the first of its kind in terms of scale and propagation rate, taking advantage of the flaws of the old systems and encrypting all data as well as requiring money in the form of cryptocurrency [4]. The outbreak in 2017 affected the healthcare services, financial institutions, and logistics networks in over 150 countries, showing the vulnerability of the world to advanced malware attacks [5]. Since, ransomware landscape has only diversified with attackers now using polymorphic and evasive methods to counter traditional security measures [6]. Simultaneously, the scale and the sophistication of the malware samples have increased and proper classification has become more and more challenging [7]. The available detection systems are mostly built on a static or signature-based approach and do not generalize with the changing ransomware families or differentiate between a particular attack such as the WannaCry and a larger malware ecosystem [8, 9].

In order to deal with such challenges, the community of cybersecurity has resorted to deep learning as it has the ability to extract features automatically and recognize patterns [10-13]. Nevertheless, there are two major limitations of most of the modern deep learning methods of detecting malware [14]. First, they commonly detect raw or engineered features without choosing the most salient features, and this can lead to either overfitting or poor performance on high-dimensional, noisy data [15]. Second, most architectures do not offer the interpretability necessary so that security analysts can comprehend and trust the results of the detection process - which is a necessary condition in high-stakes decision making scenarios. However, there is still an urgent necessity to have those models that are not only precise and generalizable but also transparent and sound in tasks that require variant-specific detections [16].

We introduce DeepRansom, a new deep learning system specific to WannaCry ransomware detection among malware families in this paper. The general concept behind DeepRansom is that it is a dual-model framework combining both the advantages of two state-of-the-art components: TabNet, a deep tabular learning model with inherent feature selection and FT-Transformer, a transformer-based network tailored to tabular data. TabNet allows the model to specialize in the most informative aspects of the data, eliminating noise and enhancing interpretability, and the FT-Transformer allows complex, non-linear features-to-features interactions to be learned by attention. It is this synergy that enables DeepRansom to perform not only with high precision in the detection of WannaCry, but also to be resilient to false positives due to other types of malwares even under conditions of class imbalance or new samples.

The main findings of the study are as follows:

1. We present a hybrid deep learning model, which uses TabNet to select the features and the representational capabilities of FT-Transformers to detect ransomware.
2. We show that DeepRansom is more effective in detecting WannaCry compared to the traditional and state-of-the-art baselines in heterogeneous malware data.
3. To enable actionable insights to threat analysts, our visual analytics of learned feature importance's with SHAP method offer interpretability of these features. The remainder of this paper is organized as

follows: Section 2 reviews related work in malware and ransomware detection using deep learning. Section 3 presents the design of the DeepRansom framework, detailing the feature selection and FT-Transformer modules, describes the dataset, preprocessing steps. Section 4 reports the results, including comparative evaluations and interpretability analyses. Finally, Section 5 concludes the paper with a summary of findings, discusses limitations, practical implications, and future work.

2. Literature Review

The increased complexity and variety of malware, especially ransomware, have prompted a mass of research into computer-aided detection protocols [17]. Classical methods largely depend on signature-based methods of detection, which determines threats through known signatures in malicious code. Although it works well with problems already faced. malware, these approaches fail to face new or obfuscated versions, like WannaCry, because of their relied on existing signatures [4, 18].

In an attempt to overcome these shortcomings, heuristic-based methods of detection have been invented. Such approaches examine code behavioral and features to detect suspicious behavior, which facilitates detection of hitherto unknown threats [19]. Nevertheless, the heuristic methods tend to have a high false-positive and might be not accurate in terms of results malware classification.

With the introduction of machine learning (ML), more adaptive detection mechanisms have been suggested. Malware classifiers such as Support Vector Machines (SVM), Random Forests, and k-Nearest Neighbors, have been used to classify malware based on the features extracted out of the binaries, API calls, or execution traces. Although these models have better flexibility compared to rule-based systems, their functionality can be rather limited by quality of manually engineering features and can be not transparent in decision-making procedures [20, 21].

Malware discovery Deep learning (DL) methods have also transformed malware discovery by enabling models to learn intricate patterns directly out of the data. CNNs have been applied to model sequence of bytes and API call traces, and they have been found to be quite successful in classification of malware activity [22] and RNNs [23]. Transformer-based architecture has also shown promise in the nearer past because of its attention mechanisms and were shown to capture long-range dependencies in data. As an example, MalConv uses raw sequences of the executable files in order to detect malware [24]. Equally, ConvLSTM models have been demonstrated to be effective in detecting fileless malware through the addition of temporal dependencies in the system behavior [25, 26].

Nevertheless, there are still challenges as a result of these developments. Most DL models have been trained on generic malware and are not necessarily very effective at isolating individual members of the variants such as WannaCry when presented in a mixed malware set. Also, the interpretability of the DL models is a challenge since most often, the complex architecture of a specific model is used as a black box and the mechanism by which the model makes decisions is poorly understood. This non-transparency may negatively affect their implementation in security critical settings where it is necessary to know the reason behind the detections [27].

To solve such problems, recent research has considered the use of models designed using tabular data, which is common in malware datasets. One such example is TabNet, which utilizes a sequential

attention mechanism to filter out the relevant features to improve the performance and interpretability of malware detection activities [28]. In the same way, the FT-Transformer adapts the transformer structure to structured data, converting features into tokens and making use of self-attention layers, letting the model discover and model complex interactions between features [29].

Based on these advances, our own framework, DeepRansom, combines the feature selection facilities of TabNet with the representational strength of the FT-Transformer. The proposed hybrid architecture is designed to increase the accuracy of WannaCry detection using heterogeneous malware samples and give interpretable information concerning how the model decides. To the best of our knowledge, this is among the first studies to combine TabNet and FT-Transformer for targeted WannaCry detection within a heterogeneous malware environment.

3. Materials and Methods

To get a high detection rate of WannaCry ransomware, weused a two-model deep learning pipeline, i.e., TabNet and the FT-Transformer. TabNet: TabNet is a tabular data DL model, which employs sequential attention to select useful features in each decision step. Since the model includes its built-in capability of selecting the most informative features, it is better able to improve interpretability and reduce overfitting.

FT-Transformer: FT-Transformer is a self-attention-based transformer which is specially designed to work with tabular data. Drawing enhanced-order dependencies by the features, the FT-Transformer supplements the feature selection of TabNet, which allows the integrated model to acquire subtler patterns that reflect the presence of WannaCry ransomware.

3.1. Dataset Description

In this research, we have used Ransomware Dataset 2024 which is publicly accessible on Zenodo. This rich data is carefully selected to analyze malware and is a total of 21,752 samples; 10,876 of the samples are malware files and 10,876 are benign files, well-balanced to be used in machine learning. The malicious samples consist of 26 different malware families, the dominant one being ransomware. It is important to note that the dataset contains 11 most prevalent ransomware families: Cerber, Darkside, Dharma, GandCrab, Lockbit, Maze, Phobos, Revil, Ragnar Locker, Ryuk, Shade and WannaCry. To ensure clarity, the dataset composition is summarized as follows. The total dataset contains 21,752 samples, including 10,876 benign files and 10,876 malicious files. Among the malicious samples, 26 malware families are represented, including 11 ransomware families. Within this dataset, WannaCry samples constitute the positive class in our binary classification formulation, while all other ransomware families and non-ransomware malware samples are grouped into the negative class along with benign files. All malicious samples correspond to executable files collected from real-world malware repositories. The dataset includes both static and behavioural features extracted from these executables. Such families are among the most infamous ransomware types that caused major cyberattacks over the past few years.

The dataset is divided into four major types, but the criteria of the specific categorization are not described. Common classifications of malware data sets consist of classifications like Trojan, Ransomware, Spyware and Adware. This dataset is especially fit to be used in the training and evaluation of ML models that can detect malware due to its well-balanced character and variety identification and categorization.

3.2. Problem Formulation

The main objective of this research is to suggest a strong deep learning model that is able to precisely identify WannaCry ransomware and other malware groups and benign software. It is formulated as a binary classification problem, and each of the samples is described as warranting labeling as either "WannaCry" (positive class) or as Other (negative one). The trick is in the ability to identify the peculiarities of the WannaCry ransomware and, at the same time, assure the model can be generalized to a variety of malware representations and harmless files.

3.3. Data Preprocessing

Prior to the training of our hybrid deep learning model, there were a number of data preprocessing steps performed so as to maintain the quality and compatibility of the data with the learning algorithms. The dataset included 21,752 samples and there was an equal balance of less than and more than. There were 10,876 benign and 10,876 malicious files, which allowed setting a balanced classification environment. A was abundantly annotated in each sample large number of both static and behavioral characteristics, such as file headers, PE (Portable Executable) metadata, API call, statistics, and indicators of registry/network/process/file activity.

To ensure there was no redundancy or corruption of data, first, we had to remove the duplicates and ensure data integrity. Since most of the initial fields, like the values of hexadecimal PE structure, were in the form of strings, we interpreted and converted the relevant fields to other types using the relevant parsing methods (e.g., the conversion of hexadecimal string to integers). This was an essential step in order to guarantee that the deep learning models would be able to act in a consistent and useful feature space.

Then, feature selection and encoding were performed. Label-encoded such categorical features as file_extension, PEType, and MachineType so that these features are represented as integer values, but do not lose the categorical differentiation. To the numerical characteristics with a large variance (e.g., apis, SizeOfImage, total_processes), we applied z-score normalization to bring the values to the same level to achieve convergence in the models and minimize the effects of outliers.

The labels on the targets were built by including the Family column in a binary classification system: samples that were labeled as WannaCry would be included as the positive class, and all other families, including benign samples and non-WannaCry malware, were included in the negative class.

This redefinition transformed the initial multi-class malware identification problem into a narrow binary classification problem, whose goal was to identify WannaCry ransomware, in particular. An 80-20 split ratio was used to randomly split the data into training and test sets with most of the samples being used to model training and the remaining part of the sample was used to evaluate the model. Both sets maintained the same set of classes by means of stratified sampling so that the WannaCry would be represented consistently in both training and evaluation.

3.4. Proposed Model

To be able to identify WannaCry ransomware in a highly diverse and heterogeneous malware ecosystem, we suggest a hybrid deep learning structure, dubbed DeepRansom, that combines the advantages of TabNet and FT-Transformer. The proposed dual-model pipeline will focus on resolving the main encounters of feature redundancy, interpretability and context-sensitive learning in high-dimensional tabular malware data.

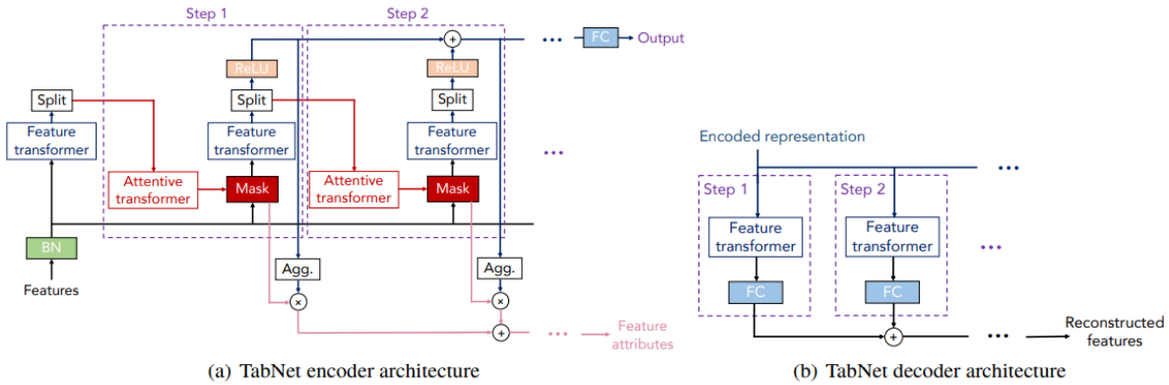

Figure 1- *TabNet architecture showing (a) the encoder with feature transformer, attentive transformer, and feature masking across decision steps, and (b) the decoder that reconstructs features from the encoded representation.*

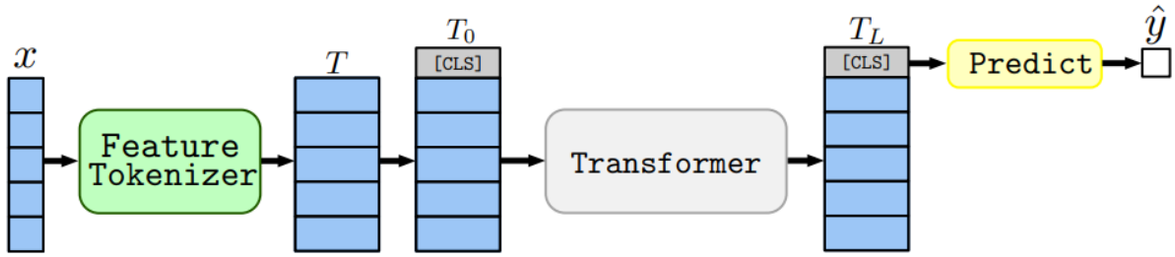

Figure 2- *FT-Transformer architecture illustrating feature tokenization, multi-head self-attention blocks, and prediction head for tabular data classification.*

TabNet is a recent deep learning model that is targeted at tabular data. Contrary to the conventional multilayer perceptrons (MLPs) where all input features are considered as possible, TabNet uses a sequential attention model which selectively chooses a smaller number of the most informative features at each stage of decision making. This allows the sparse selection of localized features conditioned on the past step context, providing sparse interpretable representation of the data.

In its simplest implementation TabNet is a combination of two fundamental elements: a feature transformer and an attentive transformer, Table 1. The input data is processed by the feature transformer with the help of fully connected layers and with the application of the batch normalization feature, which consequently generates contextualized embeddings that reflect the relation of features [30]. That is followed by the attention transformer, an activation based on the softmax alternative which is differentiable to produce masks that represent the most informative features to each sample at every step of the decision process as indicated in Figure 1.

These masks facilitate future changes in the network in such a way that only salient features contribute to the learning process. Not only does this mechanism eliminate overfitting by disregarding the noisy or irrelevant dimensions, but it can also provide transparency as it is possible to visualize the impact of each feature in each instance. In our pipeline, TabNet acts as a front-end module that filters and transforms the raw input space into a reduced, information-rich representation, optimized for ransomware-specific signals, including PE structure metadata, API patterns, and registry/network behaviors.

Following the TabNet feature selection, the transformed embeddings are passed to the FT-Transformer Figure 2, a transformer-based model

Table 1- *Implementation details and hyperparameters of the proposed DeepRansom framework*

Component	Parameter	Description
Data Split	Train/Test Split	80% / 20% (Stratified)
	Random Seed	42
TabNet	Number of Decision Steps	5
	Relaxation Factor (γ)	1.5
	Sparsity Coefficient (λ)	1e-5
	Feature Dimension	64
FT-Transformer	Number of Layers	3
	Number of Attention Heads	8
	Embedding Dimension	64
	Feedforward Dimension	128
Training Setup	Batch Size	256
	Learning Rate	0.001
	Optimizer	Adam
	Loss Function	Binary Cross-Entropy
	Early Stopping	Patience = 10 epochs
	Number of Epochs	50
Hardware	Processor	Intel Core i7 / Equivalent
	GPU	NVIDIA GPU (e.g., RTX 3060)
	RAM	16 GB
Software	Programming Language	Python 3.x
	Deep Learning Framework	PyTorch / TensorFlow
	Libraries	NumPy, Pandas, Scikit-learn, SHAP

adapted for tabular learning tasks. The FT-Transformer adapts the core components of the original transformer architecture, namely multi-head self-attention and position-wise feed-forward layers for handling tabular data by incorporating learnable token embeddings for each feature [31]. These feature embeddings are then processed through stacked layers of multi-head self-attention blocks, which allow the model to capture inter-feature dependencies, including both linear and nonlinear interactions [32]. By attending over all features in parallel, the FT-Transformer is capable of modeling complex patterns that span across feature types and malware behaviors a critical requirement in distinguishing subtle behavioral differences between WannaCry and other ransomware. Moreover, the FT-Transformer includes residual connections and layer normalization, ensuring stable and efficient training even on deep architectures. The use of shared projection weights across attention heads also reduces the number of learnable parameters, making it computationally efficient and suitable for mid-sized tabular datasets such as ours.

The DeepRansom pipeline integrates these two components into a serial configuration. First, the

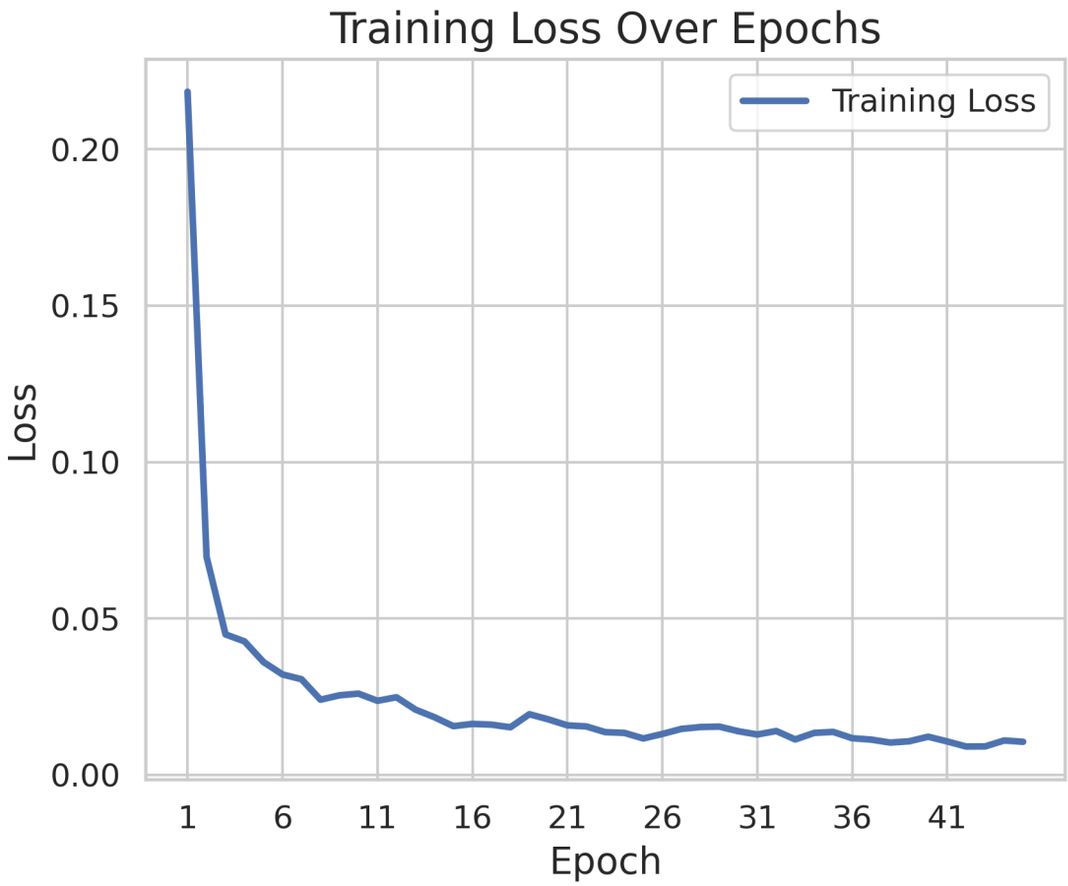

Figure 3- *Training loss curve of the DeepRansom model over 50 epochs, showing rapid convergence and stabilization at a low loss value.*

input features are passed through TabNet, which generates a contextually filtered feature subset emphasizing the most important malware attributes. These features are then fed into the FT-Transformer, which models higher-order dependencies and produces the final class logit used for binary classification (WannaCry vs. Other). The training objective for the entire model is the Binary Cross-Entropy Loss, optimized via the Adam optimizer. Early stopping is employed based on validation loss to prevent overfitting, and class weights are balanced to address the relatively small number of WannaCry samples compared to the rest of the dataset. This dual-model design brings together interpretability, feature sparsity, and context-aware learning, making DeepRansom both highly effective and transparent. The hybrid approach not only improves the robustness of WannaCry detection but also supports feature attribution analysis, which is vital for actionable insights in cybersecurity applications.

3.5. Model Evaluation

To assess the effectiveness of the proposed DeepRansom framework in detecting WannaCry ransomware, we employed a series of standard binary classification metrics. These metrics provide insight into the model's predictive performance across both the majority (benign and other malware) and minority (WannaCry) classes. Evaluation

was conducted on a hold-out test set comprising 20% of the total samples, ensuring an unbiased estimation of the model's generalization ability. Let the confusion matrix be defined as follows:

- True Positives (TP): The count of WannaCry instances that are accurately identified as WannaCry by the model.
- False Positives (FP): The count of benign or other malware samples incorrectly identified as WannaCry.
- True Negatives (TN): The count of benign samples correctly identified as non-WannaCry.
- False Negatives (FN): The count of WannaCry samples incorrectly identified as benign or other malware.

The following evaluation metrics are computed:

- **Accuracy:** Accuracy represents the percentage of correctly predicted instances (including both true positives and true negatives) relative to the total number of data points evaluated.

$$Accuracy = (TP + TN) / (TP + FP + FN + TN) \quad (1)$$

- **Precision:** Precision (commonly referred to as Positive Predictive Value) measures the ratio of correctly identified WannaCry instances to the total number of samples classified as

WannaCry by the model.

$$Precision = \frac{(TP)}{(TP+FP)} \quad (2)$$

- **Recall**: Recall (also known as True Positive Rate or Sensitivity) measures the proportion of actual

WannaCry samples that were correctly identified.

$$\textbf{Recall} = \frac{(TP)}{(TP+FP)} \quad (3)$$

- **F1-Score:** The F1 score is the harmonic means of precision and recall, offering a balanced metric especially useful when class distribution is uneven.

$$F1\text{-}Measure = \frac{(2 \times Precision \times Recall)}{(Precision+Recall)} \quad (4)$$

4. Results and Discussions

This part gives the empirical results of the proposed DeepRansom framework to identify precise WannaCry ransomware. We document convergence behavior training, metric of classification and report on the efficacy of the model in a cybersecurity setting.

4.1. Model Training

The training of the model was done over 50 epochs through the Adam optimizer with binary cross-entropy loss. To eliminate overfitting, the early stopping was employed and (the training progress is shown in Figure 3). The loss of training diminished quickly in the first epoch, showing that the training was effective and convergent. The loss had already dropped to less than 0.05 by epoch 10 and it remained stable at 0.01 in later epochs, Figure 3. This indicates that the generalization capability of this model was not overfitting, which has also been confirmed by its performance on the hold-out test set.

4.2. Quantitative Evaluation and Performance Interpretation

The proposed DeepRansom model was strictly tested using various performance indicators to fully test the efficacy of the model, such as convergence behavior, accuracy in classification, and discriminative power. The overall findings indicate the strength, accuracy, and the ability to generalize the model in identifying WannaCry ransomware.

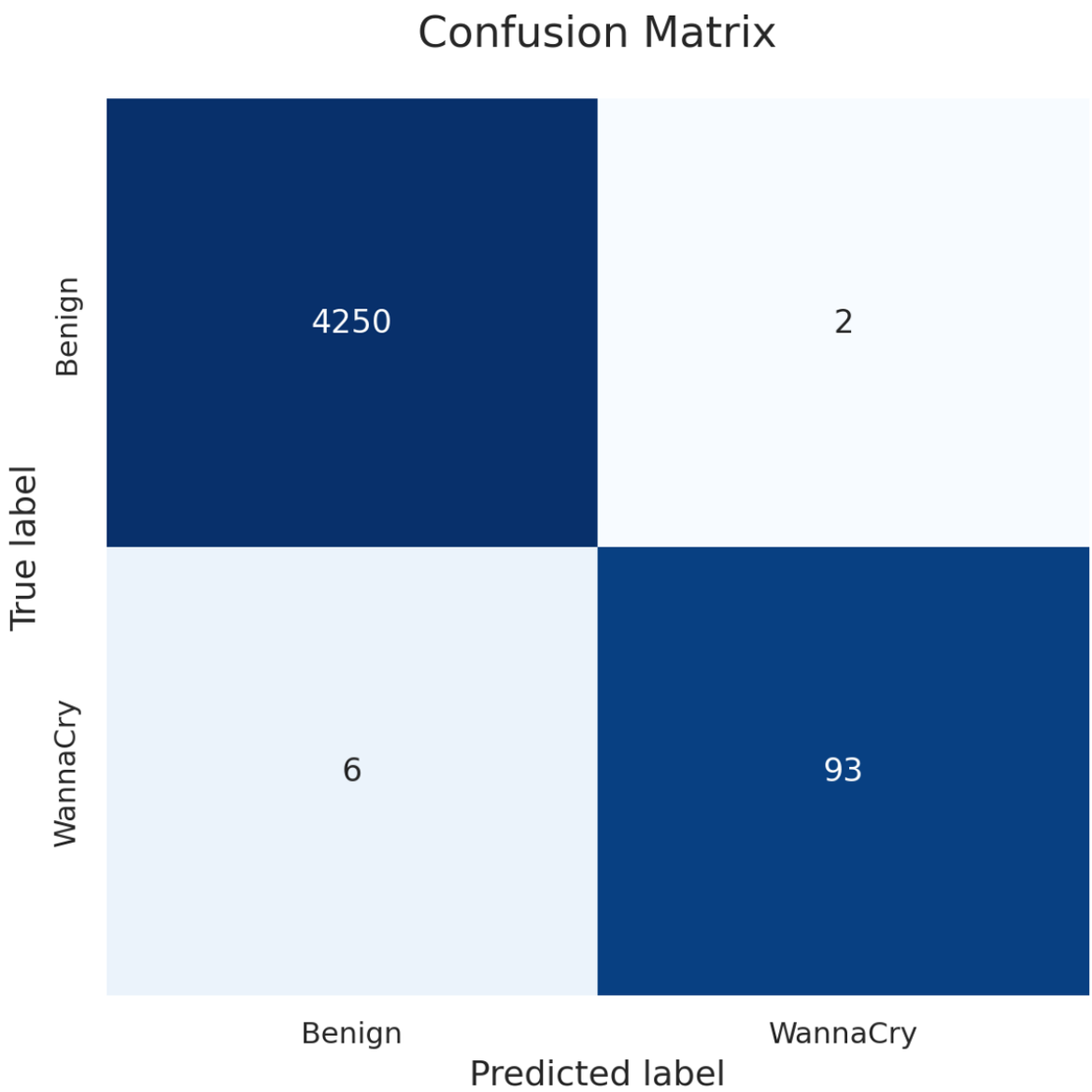

Figure 4- *Confusion matrix illustrating WannaCry versus non-WannaCry classification results on the test dataset.*

The confusion matrix indicates that the model had a high classification accuracy, with the ability to correctly classify 93 of 99 WannaCry samples and 4250 of 4252 non WannaCry samples as shown in Figure 4. It has an accuracy of 99.82% with the values of precision, recall, and F1 score of 97.89%, 93.94%, and 95.88%, respectively. The findings imply that

DeepRansom is capable of not just identifying genuine ransomware but also reducing the false positives which is a crucial parameter in any cybersecurity program since a false alarm can mean an unwarranted expenditure on mitigation measures or missed detection. Also, Precision-Recall (PR) curve that achieved an AUC of 0.97, which proves the reliability of the model in the strong balance between sensitivity and specificity, particularly under the circumstances of class-imbalanced quality as presented in Figure 5. The sharp curve is evidence of the fact that despite

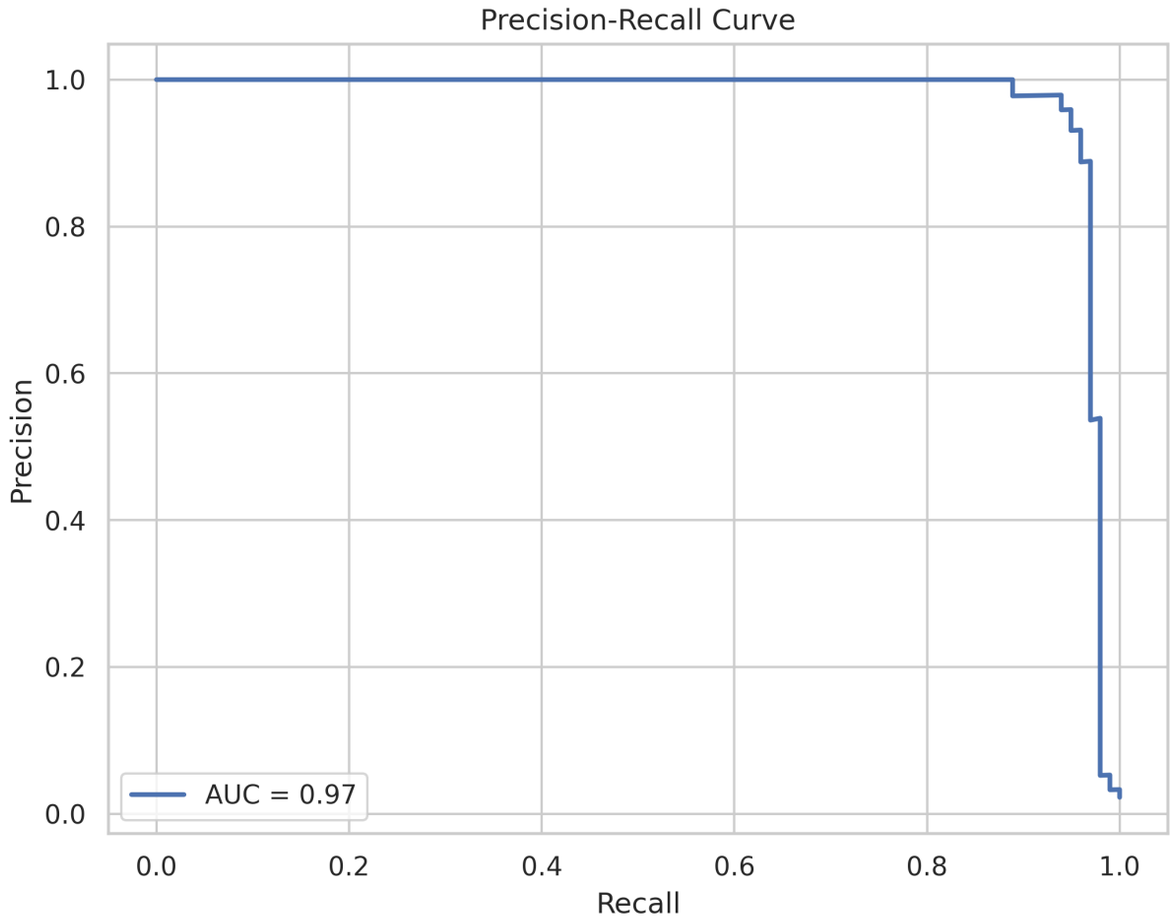

Figure 5- *Precision–Recall (PR) curve for WannaCry classification, demonstrating strong precision–recall performance under class imbalance.*

different thresholds, the model remains highly precise, minimizing chances of misclassification on real-world utilization.

To complement the PR analysis, ROC curve gave an AUC of 0.99%, which strengthens the outstanding discriminative power of the model in Figure 6. This measure is the ability of the model to distinguish between positive (WannaCry) and negative (benign or other malware) cases at a large range of classification thresholds. A combination of these results highlights how DeepRansom architecture with its dual-stage design by integrating feature-centered attention with the context-centered representation can be used to achieve high performance in terms of learning efficiency and predictive reliability. The model is operational-friendly to be used in malware detection pipelines where correct identification and interpretability is necessary equally crucial.

4.3. Model Explainability via SHAP Analysis

To enhance the interpretability of DeepRansom framework we relied on SHAP (SHapley Additive exPlanations) to determine the contribution of each feature to model predictions. The SHAP values provide both case-level information about why a particular prediction is generated and general knowledge about the overall model process of decision-making by calculating the average marginal contribution of each feature as shown in

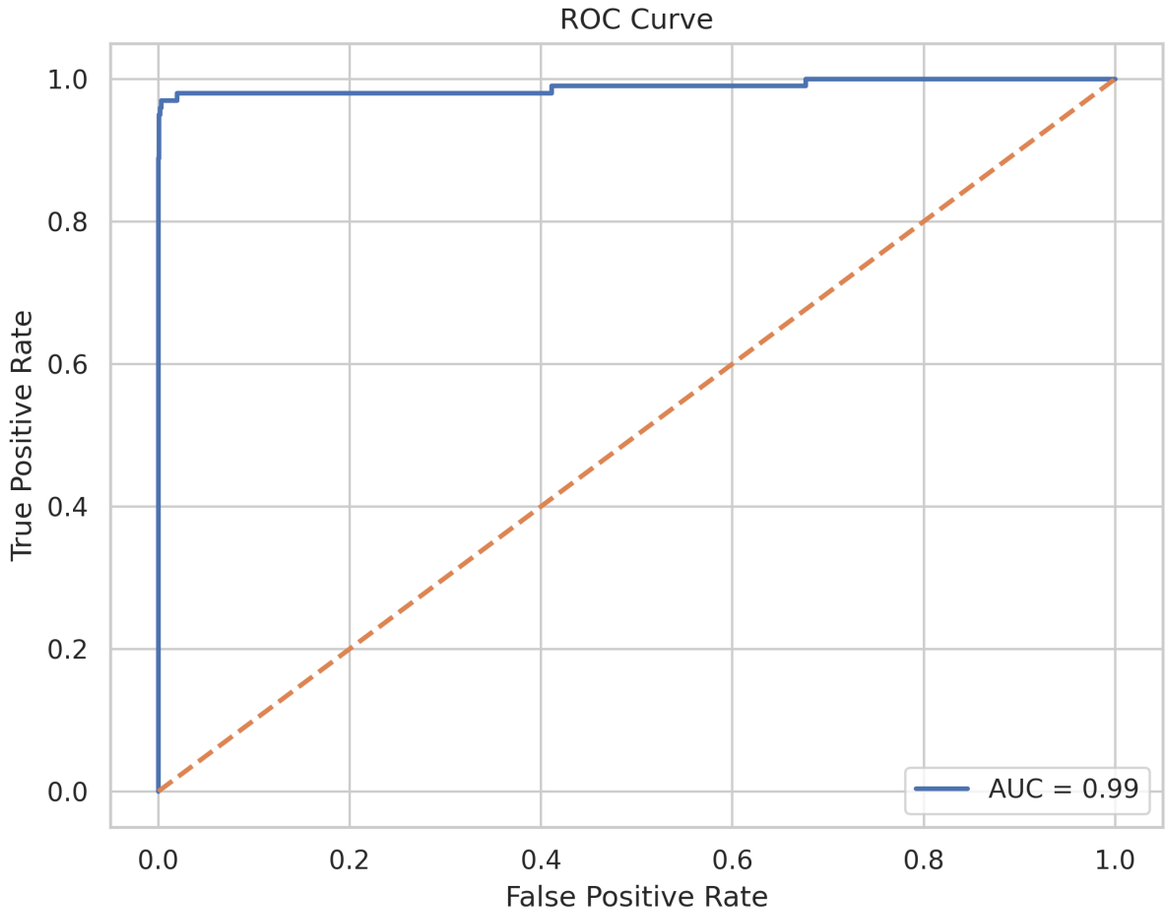

Figure 6- *Receiver Operating Characteristic (ROC) curve of the DeepRansom model showing high discriminative capability between WannaCry and non-WannaCry classes.*

Figure 7. The SHAP features ranked in terms of their importance in determining the model outputs. The value of horizontal axis indicates SHAP (the strength and direction of the impact of a feature), and the gradient of color denotes feature values (blues are associated with low value and reds are associated with high worth).

The most significant features are files_malicious and processes_malicious: High values of the indicators can greatly raise the likelihood of a sample being considered to be WannaCry. This is compatible with the previously established behavior of WannaCry as it creates malicious processes and manipulates several system files. OperatingSystemVersion: Versions of WannaCry capitalize on weaknesses of certain versions of an operating system. The SHAP contribution of the feature indicates that some of the versions are powerful discriminators. dlls_calls and files_text: These are dynamic behaviours that are usually related to the ransomware execution, including DLL injection and text-based payloads. Their large contribution confirms the relevance of behavioral aspects in detecting ransomware. SizeOfheapcommit, SizeOfstackreserve and SizeOfheaders: These are the features of static PE headers that probably encode the structural patterns that WannaCry is known to use and is more likely to do so. networkparent, registrywrite: These

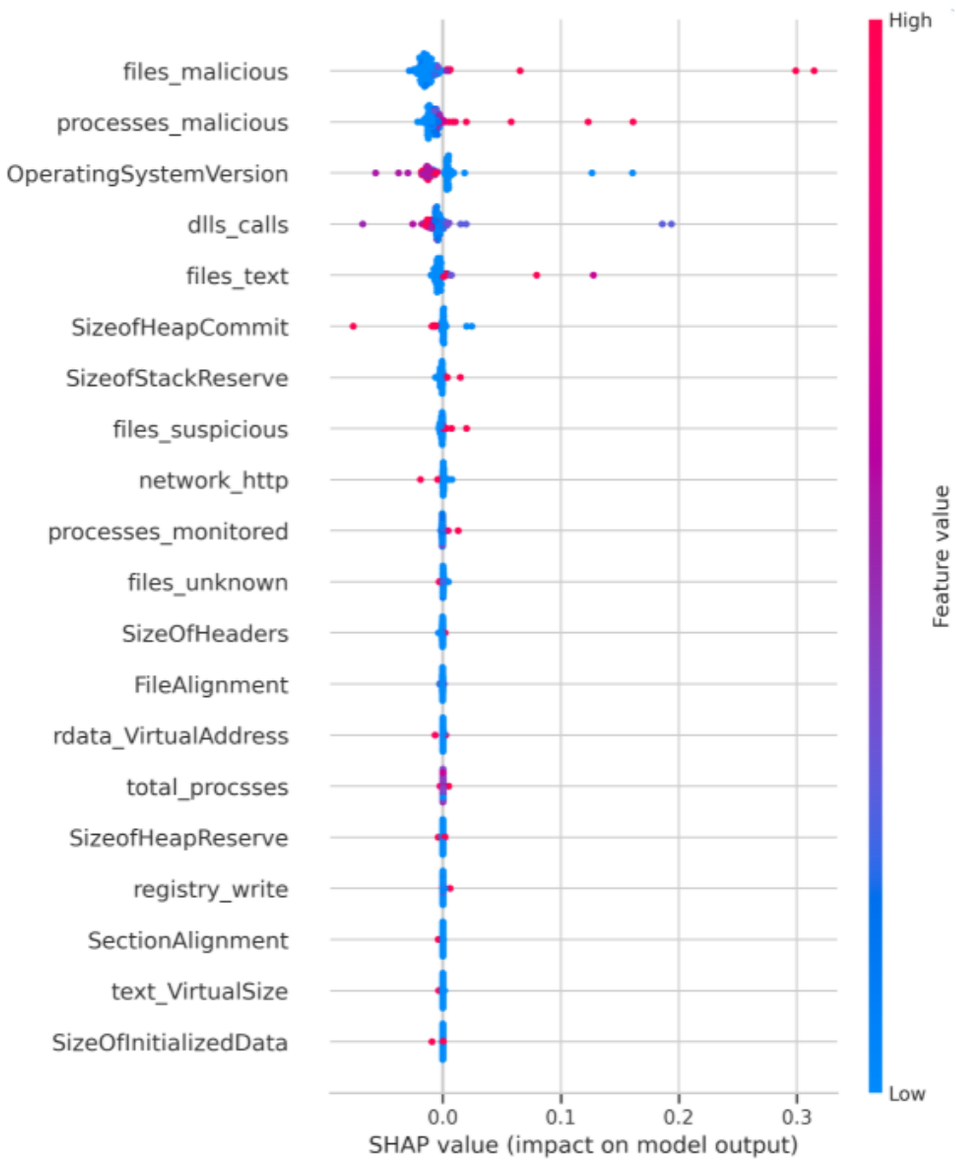

Figure 7- *SHAP summary plot illustrating global feature importance and the impact of behavioral and structural features on model predictions.*

are not the main features, but they have consistent effects, demonstrating the known communicating nature of WannaCry with commandant-control servers and their registries. The dual-model nature (i.e. the use of both behavioral attributes (e.g., file and process anomalies) and structural indicators (e.g., PE metadata) of the model prove the success of the dual-model architecture in incorporating various feature modalities. In addition, the small set of SHAP values between lower modeled features also demonstrate the sparsity that is being imposed by TabNet, which also contributes to better interpretability. Such SHAP interpretability is not only able to justify the internal logic of DeepRansom, but it also gives actionable information to security analysts, so they can see why certain samples are flagged, a critical feature in the context of trust and transparency in an operational setting.

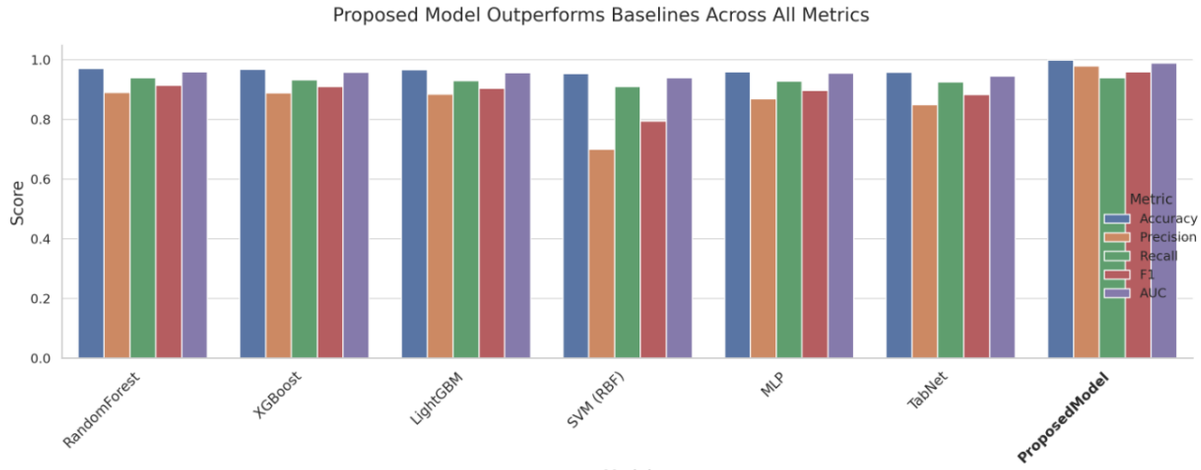

Figure 8- *Benchmark comparison of the proposed DeepRansom model against baseline machine learning and deep learning models across multiple evaluation metrics.*

4.4. Benchmarking the Proposed Model Against literature

To evaluate the effectiveness of the suggested DeepRansom, we performed a detailed benchmarking test on a wide range of baseline models, such as RF, etc. XGBoost, LightGBM, SVM with RBF kernel, MLP and standalone TabNet. The models were selected because they have been shown to be effective in the task of tabular classification and are widely used in cybersecurity and malware detection pipelines. According to Figure 8, the proposed DeepRansom model has always performed better than all other competing models based on five major performance metrics). DeepRansom has a very high accuracy of 99.82, which is much higher than even the most powerful ensemble learners, including XGBoost and LightGBM. Moreover, the model was very accurate (97.89) and recalled (93.94) cases, which depict that the model could reduce false positives as well as detect real WannaCry cases, which is an important requirement in a real threat detection context. F1-score was 95.88 which confirms that DeepRansom is a well-performing model with good performance in all classification thresholds. This AUC score value of 0.99 also demonstrates the discriminative ability of the model which has a strong capacity to differentiate WannaCry and non-WannaCry samples across the decision threshold. Although the conventional models like RF and XGBoost Classifier performed well on the baseline level, they were not rich and flexible enough to capture complex dependencies in various malware behaviors. Equally, standalone MLP and TabNet models gave reasonably good results but did not match the performance improvement that the dual-stage architecture of DeepRansom achieved by jointly combining TabNet sparse attention with FT-Transformer contextual learning. Altogether, these findings confirm clearly that DeepRansom has a serious performance benefit over existing models, due to its capacity to integrate interpretable feature selection and deep contextual representation learning. This renders it not only an effective detection engine, but also a dependable and explainable malware analysis and threat mitigation tool.

5. Conclusion

This paper has presented DeepRansom, a dual-model DL architecture that can be used to accurately detect WannaCry ransomware in a non-homogeneous environment of malware and benign

examples. Through a combination of TabNet, interpretable feature selection mechanism, and the context-sensitive learning ability of the FT-Transformer, our solution provides not only high classification accuracy but also transparency of the model, which are both important in a real-world cybersecurity system. According to experimental findings, DeepRansom is much better at performance than both classical ML models and deep learning baselines in all metrics of evaluation, which is 99.82% accuracy, 97.89% precision, 93.94% recall, and 0.99% AUC. In addition, the SHAP analysis demonstrated significant contributions of features in accordance with the knowledge about domains, which supports the credibility of the model and its possible application in malware forensics and automated threat mitigation systems.

This work is not devoid of limitations in spite of its good performance. To start with, the model is very effective at identifying WannaCry ransomware but its applicability to new ransomware families or zero-day ransomware has not been well-tested. Second, the analysis is presently confined to the static and behavioral indicators based on the known malware binaries, it may be better to add dynamic runtime monitoring indicators or network traffic patterns to enhance the detection power. Also, even though TabNet becomes more interpretable, the end-to-end architecture can be enhanced with more causal explainable techniques to provide more information to cybersecurity analysts.

The future work will be aimed at expanding DeepRansom into a multi classifier which will allow identifying and classifying a range of ransomware families not including WannaCry. We also intend to add online learning capabilities so as to respond to the threat signature changes and concept drift as time progresses. Finally, integrating DeepRansom with a live intrusion detection system (IDS) and testing it in conditions of adversarial attack will be one of the steps towards the operational deployment and hardening.

Acknowledgments

The authors would like to express their deep thanks to the Vice Presidency for Scientific Research at Naif Arab University for Security Sciences for their kind encouragement of this work.

Source of Funding

The authors received no financial support for the research, authorship or publication of this article.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Karim N. Comprehensive Analysis of Ransomware Evolution and Countermeasures in the Era of Digital Transformation. Int J Adv Cybersecurity Syst Technol Appl. 2024;8(12):20-30.
2. Irwin AS, Dawson C. Following the cyber money trail: Global challenges when investigating ransomware attacks and how regulation can help. J Money Laund Control. 2019;22(1):110-131.
3. Lang M, Connolly L, Taylor P, Corner PJ. The evolving menace of ransomware: A comparative analysis of pre-pandemic and mid-pandemic attacks. Digit Threats Res Pract. 2023;4(4):1-22.
4. Akbanov M, Vassilakis VG, Logothetis MD. WannaCry ransomware: Analysis of infection, persistence, recovery prevention and propagation mechanisms. J Telecommun Inf Technol. 2019;(1):113-124.
5. Mohurle S, Patil M. A brief study of WannaCry threat: Ransomware attack 2017. Int J Adv Res Comput Sci. 2017;8(5):1938-1940.
6. Kalinaki K. Ransomware Threat Mitigation Strategies for Protecting Critical Infrastructure Assets. In: Ransomware Evolution. CRC Press; 2025. p.120-143.
7. Gómez Hernández JA, García Teodoro P, Magán

Carrión R, Rodríguez Gómez R. Crypto-ransomware: A revision of the state of the art, advances and challenges. Electronics. 2023;12(21):4494.

8. Mazhar L, Rohatgi S. Malware Analysis and Detection: New Approaches and Techniques. In: Emerging Threats and Countermeasures in Cybersecurity. 2025. p.83-109.
9. Kang Q, Gu Y. A survey on ransomware threats: Contrasting static and dynamic analysis methods. 2023.
10. Okafor MO. Deep learning in cybersecurity: Enhancing threat detection and response. 2024.
11. Lu G, Liu Y, Chen Y, Zhang C, Gao Y, Zhong G. A comprehensive detection approach of WannaCry: principles, rules and experiments. Proc IEEE CyberC. 2020:41-49.
12. Akbanov M, Vassilakis VG, Logothetis MD. Ransomware detection and mitigation using software-defined networking: The case of WannaCry. Comput Electr Eng. 2019;76:111-121.
13. Shaukat K, Luo S, Varadharajan V. A novel deep learning-based approach for malware detection. Eng Appl Artif Intell. 2023;122:106030.
14. Aboaoja FA, Zainal A, Ghaleb FA, Al-Rimy BAS, Eisa TAE, Elnour AAH. Malware detection issues, challenges, and future directions: A survey. Appl Sci. 2022;12(17):8482.
15. Tayyab UEH, Khan FB, Durad MH, Khan A, Lee YS. A survey of the recent trends in deep learning based malware detection. J Cybersecurity Priv. 2022;2(4):800-829.
16. Rathore H, Agarwal S, Sahay SK, Sewak M. Malware detection using machine learning and deep learning. Int Conf Big Data Anal. 2018:402-411.
17. LaRocque A, Gross G, Lindholm F, Greco P, Dupont B, Kruger J. Effective ransomware detection using autonomous patternbased signature extraction. 2024.
18. Redhu A, Choudhary P, Srinivasan K, Das TK. Deep learning-powered malware detection in cyberspace: a contemporary review. Front Phys. 2024;12:1349463.
19. Stastne S, Johansson S, Laurent S, Kruger T, Fitzgerald G. Dynamic signal-based ransomware detection with temporal-pattern profiling technique. 2024.
20. Gyamfi NK, Goranin N, Ceponis D, Čenys HA. Automated system-level malware detection using machine learning: A comprehensive review. Appl Sci. 2023;13(21):11908.
21. Gorment NZ, Selamat A, Cheng LK, Krejcar O. Machine learning algorithm for malware detection: Taxonomy, current challenges, and future directions. IEEE Access. 2023;11:141045-141089.
22. Jeon S, Moon J. Malware-detection method with a convolutional recurrent neural network using opcode sequences. Inf Sci. 2020;535:1-15.
23. Halim MA, Abdullah A, Ariffin KAZ. Recurrent neural network for malware detection. Int J Adv Soft Compu Appl. 2019;11(1):43-63.
24. Khammas BM. Ransomware detection using random forest technique. ICT Express. 2020;6(4):325-331.
25. Gajdošík A, Kotuliak I. Machine Learning and Transformers for Malware Analysis: Overview. NTSP. 2024:1-7.
26. Seneviratne S, Shariffdeen R, Rasnayaka S, Kasthuriarachchi N. Self-supervised vision transformers for malware detection. IEEE Access. 2022;10:103121-103135.
27. Manthena H, Shajarian S, Kimmell J, Abdelsalam M, Khorsandroo S, Gupta M. Explainable Malware Analysis: Concepts, Approaches and Challenges. arXiv. 2024;arXiv:2409.13723.
28. Faridun R, Im EG. Enhancing Malware Detection with TabNetClassifier: A SMOTE-based Approach. KIPS Conf. 2024:294-297.
29. Praveen SP, Bikku T, Muthukumar P, Sandeep K, Sekhar JC, Pratap VK. Enhanced Intrusion Detection Using Stacked FT-Transformer Architecture. J Cybersecurity Inf Manag. 2024;13(2).
30. Somvanshi S, Das S, Javed SA, Antariksa G, Hossain A. A survey on deep tabular learning. arXiv. 2024;arXiv:2410.12034.

31. Huang X, Khetan A, Cvitkovic M, Karnin Z. TabTransformer: Tabular data modeling using contextual embeddings. arXiv. 2020;arXiv:2012.06678.

32. Ambekar NG, Devi NN, Thokchom S, Yogita. TabLSTMNet: enhancing android malware classification through integrated attention and explainable AI. Microsyst Technol. 2025;31(3):695-713.