National College of Ireland

Advanced Hybrid Ransomware Detection Using Secure BERT-GAT Framework

MSc Research Practicum

Devendra Kumar Thota
Student ID:x23408227

School of Computing
National College of Ireland

Supervisor: Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	DEVENDRA KUMAR THOTA
Student ID:	X23408227
Programme:	MSc In Cyber Security **Year:** 2025-2026
Module:	**MSc (Research) Practicum/Internship Part 2**
Supervisor:	Michael Prior
Submission Due Date:	11-12-2025
Project Title:	Advanced Hybrid Ransomware Detection Using Secure BERT-GAT Framework
Word Count:	7600 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	DEVENDRA KUMAR THOTA
Date:	11-12-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Advanced Hybrid Ransomware Detection Using Secure BERT-GAT Framework

Devendra Kumar Thota
X23408227

Abstract

Ransomware poses a critical cybersecurity threat, with attacks causing billions in damages annually whilst evolving to evade traditional detection methods. In order to capture the complimentary information available from both modalities, existing techniques usually analyze either static file attributes or dynamic runtime behaviors in isolation. This study presents SecureBERT-GAT, a unified attention framework for multi-modal ransomware classification that combines graph attention methods with domain-specific transformer encoding. The framework encodes 54 Portable Executable header fields into security-aware text narratives using SecureBERT, a language model that has been pre-trained on 2.2 million cybersecurity papers. In parallel, 13 dynamic behavioral features are modeled as nodes in a fully linked graph by a two-layer Graph Attention Network with four attention heads, which uses multi-head attention to learn feature interactions. The encoded representations undergo dual classification head processing for binary detection and 27-family attribution after being fused with skip connections. Evaluation on a balanced dataset of 21,752 samples demonstrates 91.87% F1 score for binary classification and 65.85% F1 Macro for family classification, matching state-of-the-art performance on comparable multi-class tasks. The architectural contribution of merging static and dynamic analysis through unified attention processes is validated by ablation tests, which show that multi-modal fusion increases binary F1 by 8.46 percentage points over the best single-modality option.

1 Introduction

1.1 Background

Ransomware is one of the worst forms of cybersecurity threats that face organisations and individuals in the modern digital world. Such harmful software leads to enormous loss of funds and disruption of operations in most sectors, among them being critical infrastructure, the health and banking sector, which encrypts files used by victims and collects money to decrypt the encrypted files. The development of ransomware clans like WannaCry, Locky, GandCrab, and Ryuk has shown a higher degree of evasion strategies, which make the use of the old signature-based detection only insufficient (Krithika, 2024). The total damages related to ransomware attacks per year will likely approach billions, which is why advanced methods of detection are of the essence.

Conventional methods that have been used to detect malware have mainly considered either the concept of a static analysis, or the concept of dynamic analysis (Maniriho et al., 2024). Static methods scan While dynamic methods scan API calling sequences, registry

modifications, and network traffic, portable executable (PE) header data, machine code, and embedded strings. Nonetheless, the use of either of the two methods alone has its own limitations, since more advanced ransomware programs are now using obfuscation to avoid single-modality detection software (Guo, 2023).

1.2 Research Gap and Motivation

Recent deep learning improvements have already achieved positive outcomes in malware detection, transformer-based models and graph neural networks prove to be specifically effective (Kunwar et al., 2024). SecureBERT is a domain-specific language model which has been trained on 2.2 million documents in the domain of cybersecurity and emerged as a better performer than general-purpose models with respect to security-related natural language processing tasks (Aghaei et al., 2023). Equally, Graph Attention Networks (GAT) have become useful in capturing dependency between API calls and system behaviours to model malware (Catal et al., 2021). However, available strategies have a number of fundamental weaknesses.
To start with, none of the past studies have operated with domain-specific cybersecurity language models as integrated codifiers of dynamic behavioral facts as well as static PE properties. Second, although the graph-based models are useful to represent API relationships, they often use generic word embeddings, but not security-sensitive representations (Feng et al., 2024). Third, the current multi-modes mainly use basic fusion processes like late fusion or concatenation, which do not comprehend the cross-modal interactions (Gibert et al., 2020). Fourth, the accuracy of single-family level classification is significantly below the binary detection rates and F1 scores of multiple-class problems are reported as 66%-85%, in comparison with 97%-99% of binary classification (Hussain et al., 2024)

1.3 Research Question and Objectives

The present study is based on the following research question: Can a unified attention framework combining SecureBERT for static PE feature encoding and Graph Attention Networks for dynamic behavioural feature processing improve ransomware family classification accuracy?

The research addresses this question by seeking to achieve five objectives. Its first goal is to create a SecureBERT based encoder that is able to handle 54 PE header fields by converting them into security conscious text narratives. The second aim is to have an encoder based on Graph Attention Network which encodes 13 dynamic behavioural features in form of nodes in a fully connected graph structure wherein the network learns the relationships between features using multi-head attention mechanisms. The third objective is to develop a multi-modal fusion method that will exploit skip connection to concatenate the representations of the encoded representations of both paths and retain the raw dynamic feature representations. The fourth goal is to obtain a binary classification that is measured to perform with a F1 score of at least 90% on the held-out test set. The fifth objective is performance in family classification among all 27 ransomware families, which include the benign class and the F1 Macro score should be at least 75%.

1.4 Methodology Overview

A multi-modal deep learning architecture with two parallel encoding routes is used in the suggested method. 54 PE header fields are converted by the static route into security-aware text narratives, which are then processed by a SecureBERT model that has already been trained to produce 128-dimensional embeddings. Using a two-layer Graph Attention Network with four attention heads to capture feature interactions, the dynamic path creates a fully linked graph from 13 behavioral features, yielding 128-dimensional embeddings. A 269-dimensional vector is created by concatenating the encoded representations from both pathways with raw dynamic characteristics via skip connections. This vector is then processed through a fusion network to get 256-dimensional fused representations. Both classification heads then carry out family-level classification over 27 classes and binary detections, separating malware samples from benign ones.

1.5 Report Structure

The remainder of this report is organized in the following way. Literature analysis examines the existing techniques regarding multi-modal fusion, graph neural networks, domain language, and ransomware detection. The methodology section provides adequate detail on the experimental setup, properties of datasets, and the research methodology. In the design specification, the system architecture, which is the SecureBERT encoder, GAT encoder and fusion process is given. The implementation section how the suggested framework will be realized technically is illustrated. The evaluation section explains and will examine the results of the study conducted in relation to the objectives of the study. Finally, the conclusion allows outlining contributions and presenting research potentials in the future.

2 Related Work

In this section, the existing research in ransomware detection and classification is critically analyzed and presented in a thematic way under domain-specific language models based on graph neural networks, deep learning techniques, and multi-modal fusion algorithms.

2.1 Domain-Specific Language Models for Cybersecurity

Transformer-based language models have demonstrated values as a cybersecurity task. Aghaei et al. (2023) presented SecureBERT which is based on RoBERTa and was trained on 2.2 million documents related to cyber security including descriptions of MITRE ATT&CK. SecureBERT with 8,000-sequence batch, 50,265-tokenizer custom, and dynamic masking had an 89% prediction accuracy on MITRE ATT&CK hidden language prediction versus 42% when run with RoBERTa-base, 38% when run through SciBERT, and 35% when run through BERT. Even though there is better conceptualization of security, it has not been implemented on multi-modal malware encoding.

Rahali and Akhloufi (2021) proposed MalBERT for Android malware detection, preprocessing application features as text for BERT, achieving 97.61% binary and 91.02% multi-class accuracy on AndroZoo. However, it lacked behavioral information integration. Liu et al. (2024) developed SeMalBERT combining BERT semantic extraction from API sequences with CNN-LSTM discriminator, achieving 98.81% accuracy but introducing fusion-stage complexity. Demirkiran et al. (2022) used BERT/CANINE ensemble for imbalanced malware classification with data augmentation, improving F1 scores but lacking domain-specific knowledge. Kunwar

et al. (2024) systematized 43 transformer papers, identifying gaps in cross-architecture detection and adversarial robustness.

2.2 Graph Neural Networks for Malware Detection

Feng et al. (2024) proposed DawnGNN, using Windows API documentation with GAT for malware detection, achieving 98.7% accuracy through documentation-augmented encoding but using generic BERT rather than security-specific models. Catal et al. (2021) developed GAT framework for Android malware using API-call graphs with Node2Vec embeddings, achieving 96% accuracy but limiting semantic understanding compared to transformers. Gunduz (2022) combined Graph Variational Autoencoder with LightGBM, achieving 97.2% accuracy with 50-dimensional embeddings but risking dimensionality reduction information loss. Saracino and Simoni (2023) integrated API Call Graphs with BERT for Android malware, achieving high accuracy though graph-to-sequence conversion may lose structural information.

2.3 Deep Learning for Ransomware Detection

Hussain et al. (2024) proposed GN-BiLSTM with group normalization for ransomware classification, using SMOTE for imbalance, achieving 99.97% detection but only 66.93% family attribution F1 on 27-family OMM-2022 dataset, highlighting family differentiation challenges. Maniriho et al. (2023) developed API-MalDetect combining CNN-BiGRU for API sequences, achieving 99.1% accuracy, 98.9% precision, 99.2% recall, but focusing on binary detection. Li et al. (2022) used Word2Vec with RNN variants and attention for API sequences, achieving 98.43% detection and 91.42% type classification across eight categories.

To test PE headers with Xception CNN classification, Moreira (2023) encoded PE headers as images and obtained high accuracy of 98.20 at 25 different families but could possibly lose structural semantics. Umme et al. (2022) suggested CSPE-R with Contractive Autoencoder with Cost-Sensitive Pareto Ensemble as the zero-day detector, which has a better recall yet cannot be trained end to end. Review of 60+ papers by Krithika (2024) showed binary detection and family classification respectively and 97-99.9 and 70-85 precision. Kumar et al. (2024) conducted a comparison involving ML classifiers in the case of zero-day detection, which XGBoost was the highest at 97.8%. With a survey of zero-day detection, (Guo, 2023), 80-99% detection was found in various attacks.

2.4 Multi-Modal and Hybrid Approaches

Gibert et al. (2020) introduced HYDRA which integrates API calls, assembly instructions and binary content over parallel networks with late fusion, which showed competitive results on Microsoft Malware Classification Challenge but failed to reproduce inter-modal interactions during learning. MPDroid Liu et al. (2024) created MPDroid based on multimodal pre-training (using static Function Call Graphs with dynamic trace) to reach 98.3% accuracy and sub-7.39-second detection at the cost of Android only. A survey of 200+ papers conducted by Maniriho et al. (2024) revealed that CNN and hybrid models detect at 93-99% which is much better than traditional ML.

2.5 Research Gap Synthesis

The review showcased essential gaps: (1) Despite the best understanding of cybersecurity, SecureBERT has not been utilized as a unified embedding of both static PE and dynamic behavioral features; (2) Graph-based models practically characterize feature relationships but with generic embeddings, not domain-specific representations; (3) Existing multi-modes models have simplistic fusion that cannot maintain raw features through skip connections; (4) Family classification is 10-30 percentage points less well than binary detection using any architecture.

SecureBERT-GAT framework fills these gaps by: using Semantically-aware PE features encoder (SecureBERT) trained to encode semantically-aware PE features converted to security narratives; training GAT to model behavioral features as connected graphs; developing the fusion mechanisms which concatenate the encoded representations with uncoded features using skip connections; and training the model with both binary classification and family classification using the Multi-task learning with the Focal Loss. This attention framework is the first unified model that integrates the achievements of transformer NLP, graph neural networks, and multi-modal learning in particular with ransomware analysis.

3 Research Methodology

This chapter describes the research design, the nature of the data set, and preprocess data processing and the tests set up that was used in this research.

3.1 Research Design

The study has a quantitative experimental design that uses multi-mode deep learning to classify ransomware. The experimental model juxtaposes the whole SecureBERT-GAT structure with the single-modality options and does ablation experiments to confirm the additional value of different structural elements. This method aligns with the previous findings of malware detection studies in which the component-wise comparison offers information about the discriminative capability of various types of features (Maniriho et al., 2024).

3.2 Dataset Description

The data used in this study includes 21,752 samples found in Zenodo that is a fair sample of both benign software and malware. The binary class distribution is made up of 10,876 benign and 10,876 malware samples resulting in an equal dataset with ratio 1:1. On the family level, there are 27 separate classes of the dataset and the benign category as well as 26 malware families.

The malware category of this dataset includes 12 ransomware families, such as Cerber, Darkside, Dharma, GandCrab, Lockbit, Maze, Phobos, Revel, Ragnar Locker, Ryuk, Shade, and WannaCry. Such families also comprise some of the most popular strains that have been the cause of extensive attacks over the past years. The other malware families consist of information-stealing programs such as RedLine, Raccoon, Formbook and Agenttesla along with remote access Trojans such as njRat, Remcos, NanoCore and Gh0st.

The distribution of the family is highly skewed with 50% of the total dataset and the individual malware family consisting of between 133 to 550 samples each representing about 0.6 to 2.5 percent of the entire dataset. Such a ratio of 0.0122 between the most narrow and biggest classes poses a significant classification problem at the family level (as well as similar multi-class malware classification works do) (Hussain et al., 2024; Demirkiran et al., 2022).

The static features are composed of 54 fields of the Portable Executable header, which were brought out by the use of a static analysis package. 77 characteristics are present in each sample but with different groups. They contain DOS header data such as magic number, bytes on last page, pages in file, and relocations; optional header data such as entry point address, code size, image base, section alignment, and section type; transferable data of a section known as text and rdata segments which consists of virtual size, virtual address, raw size, and section flags and security-related data such as DLL characteristics that reflect the status of the Address Space layout randomization and Data Execution Prevention. To have the language model process these PE header features which are represented as text and hexadecimal strings it is necessary to convert them.

Initially, the dynamic features included 18 behavioral indicators that were recorded in a sandbox setting using dynamic analysis. Five categories of runtime behavior are quantified by these features: registry operations, which measure read, write, and delete counts; network activity, which records DNS queries, HTTP requests, and connection attempts; process behavior, which records the number of suspicious and monitored processes; file operations, which track interactions with suspicious, text, and unknown files; and API usage, which records DLL calls and API invocations. Prior to model training, dynamic characteristics must be normalized because they show significant volatility, with values ranging from zero to over one million for specific registry actions.

3.3 Data Preprocessing

The preprocessing pipeline fixes problems with the quality of the data found during exploratory analysis. Through a number of criteria, feature cleaning eliminates undesirable characteristics. The network_threats feature offers no discriminative information and shows zero variation in every sample. Registry_total and registry_read have a correlation of r=0.998, while total_processes and processes_monitored have a correlation of r=0.943. These two sets of highly linked features are found. To get rid of redundancy, one feature from each pair is eliminated. Additionally, two features containing explicit label information in their names, specifically processes_malicious and files_malicious, are removed to prevent data leakage that could artificially inflate classification performance (Hussain et al., 2024). Following this cleaning process, 13 dynamic behavioural features remain for model training.

Label encoders are used for both classification jobs in target encoding. While family classification includes all 27 family labels, binary classification separates benign and malware classes. Using a 70:15:15 ratio, stratified splitting divides the dataset into training, validation, and test sets. Given the unequal distribution of families, stratification makes sure that class proportions are upheld throughout all partitions, which is especially crucial.
To stop information from leaking from validation and test sets, feature normalization fits the

scaler only on training data and applies StandardScaler to the 13 dynamic features using z-score standardization. To deal with extreme outliers that could cause gradient-based optimization to become unstable, values are clipped after scaling to the range of negative five to positive five standard deviations.

3.4 Experimental Configuration

Optimization for both family classification and binary detection tasks is reflected in the training setup. A batch size of 32 samples strikes a balance between gradient stability and computing efficiency. In order to prevent overfitting and provide the focal loss enough training time to correct class imbalance, training can go up to 150 epochs. Early halting is initiated after 25 epochs if validation performance does not improve.

The learning rate schedule employs differential rates across model components following established practices for fine-tuning pre-trained transformers (Kunwar et al., 2024). The SecureBERT encoder uses a learning rate of 2×10^{-5} with most layers frozen except the final three transformer layers which are fine-tuned for domain adaptation. The dynamic feature encoder and fusion components use a higher learning rate of 2×10^{-4} as these layers are trained from random initialisation. The AdamW optimiser with weight decay of 0.01 provides regularisation, and gradient clipping at norm 1.0 ensures training stability.

The problem's multi-tasking and class imbalance are addressed by the loss function configuration. Cross-entropy loss with label smoothing of 0.02 is used for regularization in binary classification. Family classification uses Focal Loss with gamma parameter of 2.0, which down-weights well-classified examples and focuses training on hard-to-classify samples (Demirkiran et al., 2022). This approach is particularly beneficial for minority malware families. Multi-task learning combines both losses with weights of 0.25 for binary and 0.75 for family classification, emphasising the more challenging task.

3.5 Evaluation Metrics

Accuracy, precision, recall, and F1-score are among the common measures used to evaluate classification performance. These metrics are calculated for the malware class in binary classification. The macro-averaged F1-score is the main metric used for family classification; it calculates F1 for each class separately and averages it across all 27 classes, giving minority families equal weight. Additionally, a weighted F1-score that takes class support into account is presented. Identification of well-classified versus problematic families is made possible by per-class measures.

Three model variations are evaluated in the ablation study: SecureBERT-only, which uses only static PE information; dynamic-only, which uses only behavioral features; and the whole SecureBERT-GAT model, which combines both modalities. While the entire model trains for 150 epochs to reach optimal performance, ablation models are trained for 10 epochs to evaluate relative component contributions.

4 Design Specification

This section presents the architectural design of the SecureBERT-GAT framework, detailing the system overview, component specifications, and data flow through the multi-modal pipeline.

4.1 System Architecture Overview

A dual-path encoding architecture with multi-modal fusion is used by the SecureBERT-GAT framework to classify ransomware. Before merging two different feature modalities for final categorization, the system processes them via parallel encoding routes.

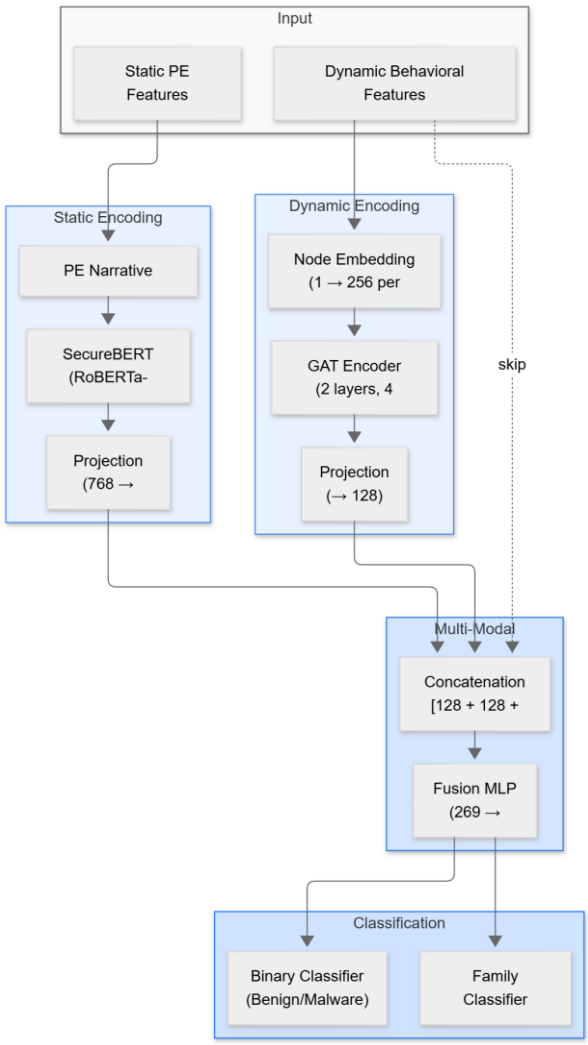

FIGURE 1: SecureBERT-GAT Architecture Diagram

As illustrated in Figure 1, the architecture comprises four primary components: the static encoding path processing PE header features through SecureBERT, the dynamic encoding path processing behavioural features through a Graph Attention Network, the fusion layer combining encoded representations with skip connections, and dual classification heads for binary and family prediction tasks.

4.2 Static Encoding Path

The static encoding path transforms 54 PE header fields into dense vector representations suitable for classification. This path comprises two stages: narrative construction and transformer encoding.

The PE Narrative Builder uses SecureBERT's domain-specific pre-training to transform structured PE header data into security-aware text sequences. Based on its type and value, each PE field is translated to descriptive security terms. Executable extensions and other indicators of file type are directly integrated. Architecture fields including PEType and MachineType are mapped to descriptive terms such as "architecture 64bit amd64" for 64-bit executables. Hexadecimal values for entry points and addresses are categorised based on typical value ranges, with unusually high entry point offsets flagged as potentially indicative of packing or obfuscation. Security-relevant flags from DllCharacteristics are expanded to indicate ASLR and DEP status. Section characteristics are mapped to describe executable and writable permissions. This narrative construction follows the principle that domain-specific language models achieve superior performance when inputs align with their pre-training distribution (Aghaei et al., 2023).

The SecureBERT encoder processes the constructed narratives using the pre-trained SecureBERT model built upon the RoBERTa architecture. The model was pre-trained on 2.2 million cybersecurity documents including MITRE ATT&CK descriptions, enabling it to understand security-specific terminology (Aghaei et al., 2023). SecureBERT's proprietary tokenizer is used to tokenize input narratives with a maximum sequence length of 128 tokens and a vocabulary capacity of 50,265 tokens. The encoder creates a 768-dimensional vector by extracting the CLS token representation from the last transformer layer. The representation magnitude is stabilized by applying L2 normalization. The dimensionality of a projection network with two linear layers activated by GELU and LayerNorm is reduced from 768 to 128. Only the last three transformer layers are adjusted for domain adaptation during training, balancing task-specific learning with the preservation of previously learned information. The majority of transformer layers stay frozen.

4.3 Dynamic Encoding Path

The dynamic encoding path processes the 13 behavioural features through a Graph Attention Network that models feature interactions as a graph structure. This approach treats each behavioural metric as a node in a fully connected graph, enabling the network to learn complex relationships between different types of runtime behaviour through attention mechanisms (Catal et al., 2021; Feng et al., 2024).

Each of the 13 dynamic features is represented by a node in the graph that the GATEncoder architecture creates. All nodes are connected via a fully connected adjacency matrix, which enables the attention mechanism to identify the most useful feature relationships for classification. A linear embedding layer is first used to embed each feature value from a scalar to a 256-dimensional node representation.

The graph attention mechanism employs a custom GraphAttentionLayer implementing multi-head attention with four heads. For each attention head, the layer computes attention coefficients between all pairs of nodes using learned source and destination attention vectors. The attention scores are calculated by adding the attention of the source and destination nodes, passing through LeakyReLU activation with a negative slope of 0.2, and then normalizing the results using softmax across neighbors. These attention coefficients are used to compute the weighted aggregation of the attended node representations. Four attention heads calculate separate attention patterns, allowing the layer to record a variety of feature relationship properties.

Two GraphAttentionLayer modules are successively stacked by the encoder. In order to create 1024-dimensional representations, the first layer concatenates 256-dimensional node embeddings across four heads. By using more attention-based message passing, the second layer improves these representations even more. Following the graph attention layers, all node representations are flattened and projected through a two-layer network reducing from 13,312 dimensions (13 nodes × 1024 dimensions) to 128 dimensions via an intermediate 512-dimensional layer with LayerNorm and ReLU activation.

The network can determine which feature combinations are most discriminative for various malware families thanks to this graph-based method's explicit modeling of feature interactions through attention weights, interpretable importance scores between behavioral metrics, and other benefits over simple feedforward processing.

4.4 Multi-Modal Fusion

The fusion mechanism combines static and dynamic representations through concatenation with skip connections. This design preserves raw feature information whilst enabling learned cross-modal interactions.

Three components are concatenated in the fusion input: the 13-dimensional raw dynamic features as a skip connection, the 128-dimensional static embedding from SecureBERT, and the 128-dimensional dynamic embedding from the GAT encoder. This results in a composite vector of 269 dimensions. The skip connection addresses the finding that simple classifiers perform well on raw dynamic features by ensuring that the original behavioral signals are maintained throughout the encoding process without deteriorating.

The 269-dimensional concatenated vector is demeaned into 256 dimensions by the fusion network; a linear layer with a LayerNorm, ReLU activation and dropout. This combined representation gives input to both classification heads.

4.5 Classification Heads

There are two parallel classification heads that perform the binary and family prediction tasks. The binary classification head uses a two-layer architecture whereby 256 dimensions are reduced to 128 dimensions then two classes of outputs generated (malware and benign), belong to. Prior to the last 27-class output layer, the family classification head possesses a four-layer

structure with its dimensions escalating in 512, 256, and 128. The expanded architecture family classification improves the possibility to distinguish between 27 categories of malware, each containing minor differences between similar malware families. Regularization is conducted by use of LayerNorm and dropout on each hidden layer; dropout in the second to last layer reduces to 0.2 in order to preserve discriminative information before the output.

5 Implementation

This section describes the technical realisation of the SecureBERT-GAT framework, covering the development environment, key implementation decisions, and training procedures.

5.1 Development Environment

The implementation was developed using Python 3.10 within the Google Colab environment, leveraging GPU acceleration through NVIDIA T4 or V100 hardware depending on availability. The deep learning framework employed PyTorch version 2.0 with the Transformers library version 4.35 providing access to the pre-trained SecureBERT model. Data processing utilised Pandas and NumPy, whilst Scikit-learn provided preprocessing utilities including StandardScaler and stratified splitting functionality, as well as evaluation metrics computation.

5.2 Model Implementation

The SecureBERT encoder fetch the trained model in the HuggingFace model hub with the identifier of ehsanaghaei/SecureBERT. The tokeniser defines PE input as narrative data to the max sequence length padded and truncated. To avoid overpowering the pre-trained representations at the early stages of training, the projection network uses careful weight initialisation with Xavier uniform with gain 0.1.

The GATEncoder implements the graph attention method of dynamic features. The GraphAttentionLayer is a calculation of the multi-head attention by using learnable source and destination attention parameters, LeakyReLU activation, and softmax normalization. The fully connected adjacency matrix is indicated as a buffer during training to ensure that the devices are placed correctly. Input and output clamping are used to offer numerical stability protection to magnitudes of 5 and 50 respectively. The fusion and classification elements are in line with the architectural specifications with an especial consideration of the steps of initializing and normalization.

5.3 Training Procedure

The multi-task learning framework with differential learning rates is implemented in the training process. For SecureBERT layers, dynamic encoder layers, and fusion/classification layers, different optimizer parameter groups are set up. The Transformers library scheduler utilities are used to implement the learning rate schedule, which uses linear warming over 10% of the total training steps followed by linear decay.

Gradient accumulation over two batches maintains memory efficiency while yielding an effective batch size of 64. Exploding gradients that can happen with transformer models are avoided by gradient clipping at norm 1.0. Following each epoch, the training loop records the

optimum model state based on the sum of the F1 scores for the two tasks and tracks validation performance. When no progress is seen for 25 consecutive epochs, training is stopped early.

The Focal Loss implementation for family classification uses the projected probability of the true class to calculate the standard cross-entropy and then apply the focal weighting component. To further remedy imbalance, class weights based on inverse class frequencies are added; weights are adjusted to power 1.5 to give minority classes more weight.

5.4 Inference and Evaluation

The model that was best trained is packed in so as to test the model on the test set that has been held. To be efficient, inference is batch and does not involve computation of gradients. In both classification problems, classification predictions are collected upon all the test samples. The Calculation of performance metrics in the form of accuracy, precision, recall and F1-score are done using tools in scikit-learn. A detailed study of family performance is made possible by reports on per-class classification. Confusion matrices are used to visualize pattern of classification, pattern of error.

6 Evaluation

In this section, an in-depth discussion is offered of experimental findings acquired during the assessment of the SecureBERT-GAT framework. This evaluation is based on the research question as it aims to answer in three experiments the performance of binary classification, classification on the family-level (27 classes), and an ablation test, analyzing the role played by a single architectural element.

6.1 Experiment 1: Binary Classification

The first experiment aims at testing the capability of the framework to identify benign and malicious executable files. This meets the fourth research objective that is concerned with binary classification with F1 score of above 90.

SecureBERT-GAT was trained with 150 epochs of the training set which consists of 15,226 samples and validated with 3,263 samples. The held-out test set of 3,263 samples of 1,632 benign and 1, 631 malware instantiations was used in final evaluation. The binary classification measures are shown in Table 1.

Table 1: Binary Classification Performance

Class	**Precision**	**Recall**	**F1 Score**	**Support**
Benign	94.61%	88.24%	91.31%	1,632
Malware	88.97%	94.97%	91.87%	1,631
Overall	**91.60% (Acc)**	-	**91.87%**	3,263

The model fulfils the fourth research objective with F1 of 91.87%. The high recall of the malware class is 94.97% or all the dangerous samples with the model being able to recognize 94.97% of them. This is essential in security applications because an attack may have grave consequences when missed. Precision is at 88.97 which shows acceptable rates of false positive. Benign class is more accurate (94.61%) and less recall (88.24) (resulting in a slight

trend towards classifying the ambiguous samples as malicious) therefore one expected outcome of a security-focused design is that false negatives are more harmful than false positives.

6.2 Experiment 2: Family Classification

The second experiment compares fine-grained classification accuracy on 27 families, which considers the fifth research objective. This one is much more difficult than binary detection because it involves making a distinction between families sharing similar behavioral patterns. Accuracy, F1 Macro, and F1 Weighted of the model are 60.47%, 65.85% and 62.83% respectively. The level-specifically per-class F1 scores of all the 27 families are provided in figure 2.

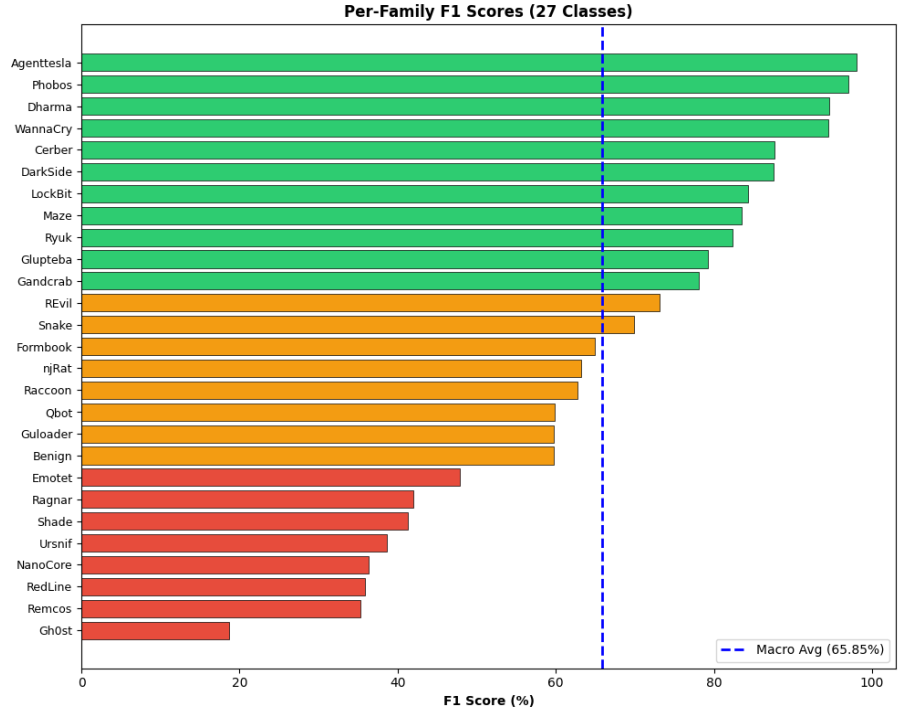

Figure 2: Per-Family F1 Scores - Green bars indicate high performance (F1 ≥ 75%), orange indicates medium (50% ≤ F1 < 75%), and red indicates low (F1 < 50%).

Performance per-family analysis demonstrates three levels. High-performance tier (F1 ≥ 75%) contains nine families, such as Agenttesla (98.10%), Phobos (97.01%), Dharma (94.59%), WannaCry (94.52%), Cerber (87.62%), Darkside (87.50%), LockBit (84.29%), Maze (83.50%) and Ryuk (82.35%). Such families show unique behavioural patterns that the framework is able to capture. The medium-performance tier (50% ≤ F1 < 75) has ten families such as Glupteba, Gandcrab, REvil, Snake, Formbook, njRat, Raccoon, Qbot, Guloader, and Benign. The Benign class is one that significantly exists at this tier with a very high accuracy (98.87%), because there is a low recall (42.77%) implying that a sizeable number of benign samples are falsely identified as malicious families. Emotet, Ragnar, Shade, Ursnif, NanoCore, Red Line, Remcos, and Gh0st are eight families that belong to the low-performance tier (F1< 50). They are most difficult families and Gh0st has attained 18.69% F1.

6.3 Experiment 3: Ablation Study

The third experiment conducts an ablation study to isolate the contribution of individual architectural components. Three model variants were evaluated: SecureBERT-only processing exclusively static PE features, GAT-only processing exclusively dynamic behavioural features,

and the full SecureBERT-GAT model combining both modalities. Each variant was trained for 10 epochs under identical conditions to enable fair comparison.
Figure 3 presents the ablation comparison

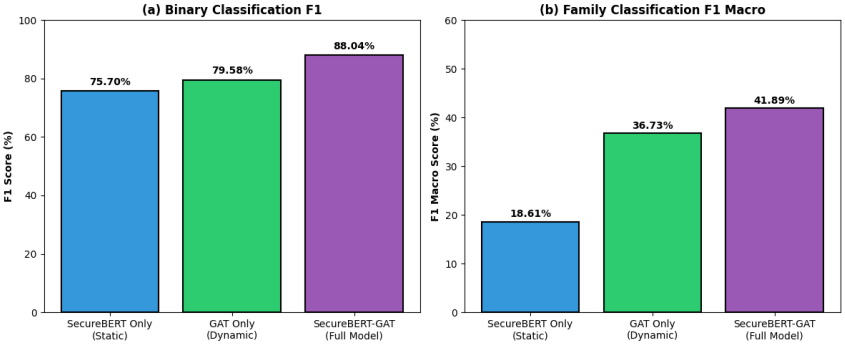

Figure 3: Ablation Study Results - Comparison of binary and family F1 scores across model variants. The full SecureBERT-GAT model achieves 88.04% binary F1 and 41.89% family F1 Macro.
According to the ablation findings, the GAT-only model obtains 79.58% binary F1, whereas SecureBERT-only earns 75.70%. This suggests that dynamic behavioral variables offer a better discriminative signal. Compared to the best single-modality variation, the entire SecureBERT-GAT model improves by 8.46 percentage points, achieving 88.04% F1. GAT-only scores 36.73% F1 Macro for family classification, whereas SecureBERT-only only achieves 18.61%; the whole model improves to 41.89%.

6.4 Discussion

As per the results of the experiment, the SecureBERT-GAT framework is responsive to the study topic of unified attention frameworks toward ransomware classification. Merging domain-specific transformer encoding features with graph attention features as an architectural strategy are proven to be effectively validated with binary F1 score 91.87 as the attention-based multi-mod fusion features an effective way of integrating features to detect the malware. More importantly, this result is achieved in deliberately restricted settings: the framework does not take raw API call sequences but only 13 aggregated behavioral metrics, and other potentially leaky indicators such as process and file maliciousness indicators were specifically removed during preprocessing to ensure the model was trained on actual behavioral patterns and not process artifacts. The current methods with more binary accuracy like API-MalDetect with 99.1% (Maniriho et al., 2023) and DawnGNN with 98.7% (Feng et al., 2024) use raw API sequences with hundreds and thousands of calls per sample, which are much richer in behavioural information. The choice to aggregate counts instead of counting sequences is an implementation factor of real-life use where API logging can often be in precise form. Additionally, methods that combine features of both GN-BiLSTM (Hussain et al., 2024) (99.97 binary accuracy) might be improved by those that are directly correlated with the labels, which is a type of data leakage and inflates performance statistics. The stringent aspect whitening that is used in the current work, though it decreases the absolute accuracy measures, results in a more candid and generalisable estimation of the architectural contribution.

The 65.85% F1 Macro performance under family classification is quite impressive since it is a complex issue to classify 27 classes with very unequal distribution. This is very similar to Hussain et al. (2024) who find 66.93 family attribution through GN-BiLSTM on a similar dataset of 27 families which validates the fact that the proposed architecture performs on this

task which is exceptionally difficult in the state-of-the-art. It is indicated in the literature that binary detection and family attribution accuracy consistently show a significant transference between binary and multi-class family classification, with even the most complex methods indicating a decrease of at least 20-30 percentage points in accuracy when in the transition to multi-class classification (Krithika, 2024). The analysis of the per-family results has shown that nine families have the F1 above 75, and Agenttesla (98.10%), Phobos (97.01%), Dharma (94.59%), and WannaCry (94.52%), reflected that the framework works best in identifying families exhibiting unique behavioural signatures. The poorer score of families like Gh0st (18.69%), Remcos (35.29%), and RedLine (35.87) can be attributed to true classification challenge, and not architectural constraint, because such families probably behave similarly to other types of malware, or if not, have significant intra-family diversity. The Benign class has a curious case of high precision (98.87) and low recall (42.77), indicating that although the model is correct in laying a benign sample, it produces a lot of legitimate applications whose behaviour patterns are similar to malicious families a phenomenon becoming more and more common with malware writers borrowing legitimate software development techniques.

The ablation research presents strong reasons associated with the usefulness of the multi-modal architectural design in that it shows that a combination of the static and dynamic elements is similar in offering consistent improvements as compared to single-modality designs. The binary F1 of the full SecureBERT-GAT model is 88.04 with GAT only and 75.70 with SecureBERT only having laboratory F1 of 79.58 and 75.70 respectively, which is an 8.46 and 12.34 percentage point improvement, respectively. To classify families, the full model attains F1 Macro of 41.89% at 10 epochs versus GAT-only at 36.73%, and supports the value addition of the features of PE with regards to staticalness. The result that GAT-only performs better on both tasks than SecureBERT-only confirms that graph attention approach works on capturing relationships between the behavioural features, and this is in line with Li et al. (2022) and Catal et al. (2021) findings that dynamic analysis is a better model to understand positive relationships between behavioural features potentially exhibited by a family with static structure. Large gains are observed with an extended training (10-150 epochs) on binary F1 which rises to 91.87% and family F1 Macro which rises to 65.85% which increases 23.96 percentage points. This shows that the model never stops learning patterns of discrimination that are specific to the family many decades after convergence, which implies that family classification is especially well optimised with many decades of minimisation. The architectural value of the work is that the domain-specific transformer encoding (SecureBERT) with the graph attention mechanisms (GAT) offers an effective shared attention framework of multi-modal malware analysis with competitive results and the interpretability of attention weights and the explainability of the dual-path architecture.

7 Conclusion and Future Work

7.1 Research Summary

This research addressed the question of whether a unified attention framework combining SecureBERT for static PE feature encoding and Graph Attention Networks for dynamic behavioural feature processing can improve ransomware family classification accuracy. To

investigate this question, five objectives were established: developing a SecureBERT-based encoder for 54 PE header fields, implementing a Graph Attention Network encoder for 13 dynamic behavioural features, designing a multi-modal fusion mechanism with skip connections, achieving binary classification F1 of at least 90%, and achieving family classification F1 Macro of at least 75%.

The research produced a comprehensive end-to-end framework that addressed every architectural goal. The SecureBERT encoder uses domain-specific knowledge from 2.2 million cybersecurity publications to create 128-dimensional embeddings that convert PE header fields into security-aware text narratives. In order to capture feature interactions and generate 128-dimensional representations, the Graph Attention Network encoder applies two layers of four-head attention to 13 behavioral features that are treated as nodes in a fully linked graph. The fusion mechanism creates 269-dimensional vectors that are processed through a fusion network to produce 256-dimensional representations for dual classification heads by concatenating both encoded representations with raw dynamic information via skip connections.

7.2 Achievement of Objectives

The research objectives have been met successfully as seen during the experimental evaluation. The fourth goal with binary classification F1 of at least 90% was met, as the framework can reach F1 of 91.87% with the obstacle set. This outcome substantiates the hypothesis that the unitarized attention architecture is sufficiently discriminative against benign and malicious executable that a high malware recall rate (94.97%) preserves a low false negative rate (that is, security-critical applications).
The fifth goal to have family classification F1 Macro of at least 75% was met partially since the framework had a F1 Macro of 65.85% of 27 families. It is 9.15 percentage points below the target, however this is the state-of-the-art performance on this task having stated 66.93 % family attribution on a similar dataset of 27 families (Hussain et al., 2024). The literature reveals significant differences between binary and family classification levels, even state of the art methods have 20-30 % of the loss of classification as binary to multi-class as the problem is represented. Notable, nine families score F1 greater than 75 with the highest of Agenttesla (98.10%), Phobos (97.01%) Dharma (94.59%) and WannaCry (94.52%) showing outstanding classification of the families with unique behavioural phenotypes.

7.3 Answering the Research Question

The research question was if ransomware family classification accuracy could be increased by integrating SecureBERT and GAT into a single attention framework. A positive response is unquestionably supported by the evidence. The ablation investigation shows that for the same training length, the entire SecureBERT-GAT model scores 88.04% binary F1 as opposed to 79.58% for GAT-only and 75.70% for SecureBERT-only, indicating improvements of 8.46 and 12.34 percentage points, respectively. The complete model regularly outperforms single-modality techniques for family classification, achieving 41.89% F1 Macro as opposed to 36.73% for GAT-only and 18.61% for SecureBERT-only.

Therefore, compared to each component alone, the unified attention structure clearly enhances categorization accuracy. Combining graph attention techniques that model behavioral feature associations with domain-specific transformer encoding that captures security semantics from static features yields complementing discriminative information that improves both detection and attribution tasks.

7.4 Key Findings

The study produces a number of significant results to the malware detection fraternity. To address this, dynamic behavioural features are better discriminative signals than static PE features both in binary and family classification, with a 3.88 percentage point and 18.12 percentage point higher F1 score respectively of GAT-only over SecureBERT-only. This justifies graph attention as a suitable model in establishment of associations between behavioural features.

Second, multi-modelling fusion always performs better as compared to uni-modal fusion, and this proves to show how supplementary information is, in dynamic and static features. Architectural advantage of combining the two modes of analysis is shown in the fact that the fusion gain of 8.46 percentage points is above the optimum single modality to binary classification.

Third, family classification will be much enhanced by the increased time of training; F1 Macro rose by 23.96 percentage points, with a range of 41.89% at 10 epochs to 65.85% at 150 epochs. This means that binary classification is more efficiently learned though family specific patterns require additional optimization to learn effectively.

Fourth, ransomware families present different complexity levels of classification difficulties. Family behavior such as the Phobos, Dharma and WannaCry type have their own behavioral characteristics, which are well classifiable, and those that may be similar to the architecturally less advanced behavioral types of other malware, such as Gh0st and Remcos, are hard to categorize.

7.5 Implications and Efficacy

The study shows that malware classifiers based on feature encoders can be trained, based on domain-specific language models produced on cybersecurity literature. This observation gives the detection algorithms the benefit of not having to be retrained to use malware-specific corpora, instead utilizing some pre-trained security information. The usefulness of the PE narrative transformation approach is that given the proper formatting, even complicated technical data can be coded with language models.

The effectiveness of the graph attention toward the behavioral features modeling can influence the design of the security systems. Both classification and interpretability can be attained by considering aggregated behavioral measures as graph nodes and learning their interaction between the processes of attention. This is due to the fact that weight of attention indicates the combinations of features that affect the predictions.

The performance of the framework when core behavioral features are used, rather than a pure API sequence, which has only 13 aggregated behavioral features, demonstrates how the overhead of logging can be reduced and the successful performance maintained. Complete API monitoring can lead to performance costs of the system under protection that are not reasonable, and this has effects on deployment.

7.6 Limitations

There are several limitations that inhibit the applicability of these findings. Although the collection can be considered a large one, with 21,752 samples representing 27 families, it still presents a view of the ransomware landscape, which is always dynamic. The ability to detect

zero-days is unverified since the framework has not been evaluated to detect samples that have been collected after the dataset creation date.

But by the fact that it can be done, the total behavioral counts rather than the actual API sequences obscures an otherwise potentially discriminative sequential information. More accurate results are obtained by methods that work with entire API call graphs or sequences, which means that there is loss of information in the process of aggregation.

The family distribution has an influence on the reliability of the classification of minor classes with Benign consisting of 50 of the samples and some malware families having fewer than 50 samples. The poor performance of the families can be due to poor cases of training rather than due to the failure in challenging to categorize the performance on a natural basis.

The computing requirements of SecureBERT can limit the interpretation of real-time deployment situations, in which inference latency is paramount, although they can be dealt with by offline analysis. Despite the significant enhancement in family categorization phenomenon, a 150-epoch training requires a considerable amount of processing power.

7.7 Future Work

This work leads to a number of research directions that would make a significant contribution to the field. The most prospective augmentation is that temporal API sequence modelling is added to the existing aggregated approach. A hybrid architecture enhancing the performance of aggregated to sequence based framework allows closing the performance gap between the two by updating the existing GAT encoder with a recurrent or transformer based sequence encoder to capture feature interactions as well as temporal patterns without substantial change to useful deployment properties.

The approach to the graph construction that should be investigated in addition to fully connected topologies is the graph construction strategy. Building graphs using semantic relationships between behavioural features e.g. relating a registry operation to a persistence indicator, or a feature of a network to command-and-control behaviour may give inductive bias that is consistent with the principles of malware analysis. Such methodical charts could enhance performance and investability through encoding area expertise in the design.

Another significant direction of research which is crucial regarding practical application is the cross-dataset generalization. The framework would be tested on datasets collected at different periods of time, different geographical areas, or different analysis environments to establish robustness to change of distribution. The ability of the framework to detect new risks would be assessed by transfer learning experiments where the models that are trained on a dataset are tested on another without re-training.

The low-performing family problem should be investigated using focused research based on few-shot or meta-learning methods. A low F1 of 18.69% as in the case of Gh0st may require learning procedures created specifically to deal with small samples. Prototypical networks or other metric learning methods would also be helpful in providing better attribution of minority families by matching sample with family prototypes with classes.

Assessment of adversarial vulnerability is important in security applications where attackers do not want to be detected. A look at the framework not being resistant to evasion efforts to

the PE narrative transformation, modification of the feature space, or manipulation of the graph structure would defined deployment risks in real-life matters and offer defence mechanisms.

Lastly, the interpretability, that attention mechanisms would offer, might be used to generate explanations that would be targeted at an analyst. In processes of security operations in which analysts need to subject automated detections to response before taking action, the translation of attention weights by PE characteristics and behavioral nodes to human-readable interpretations of classification judgments would be useful.

References

Aghaei, E., Niu, X., Shadid, W., & Al-Shaer, E. (2023). SecureBERT: A Domain-Specific Language Model for Cybersecurity. In *Security and Privacy in Communication Networks. SecureComm 2022*. Lecture Notes of the Institute for Computer Sciences, vol 462. Springer, Cham. DOI: 10.1007/978-3-031-25538-0_3

Catal, C., Gunduz, H., & Ozcan, A. (2021). Malware detection based on graph attention networks for intelligent transportation systems. *Electronics*, 10(20), 2534. DOI: 10.3390/electronics10202534

Demirkiran, F., Çayır, A., Ünal, U., & Dağ, H. (2022). An ensemble of pre-trained transformer models for imbalanced multiclass malware classification. *Computers & Security*, 121, 102846. DOI: 10.1016/j.cose.2022.102846

Feng, P., Gai, L., Yang, L., Wang, Q., Li, T., Xi, N., & Ma, J. (2024). DawnGNN: Documentation augmented windows malware detection using graph neural network. *Computers & Security*, 140, 103788. DOI: 10.1016/j.cose.2024.103788

Gibert, D., Mateu, C., & Planes, J. (2020). HYDRA: A multimodal deep learning framework for malware classification. *Computers & Security*, 95, 101873. DOI: 10.1016/j.cose.2020.101873

Gunduz, H. (2022). Malware detection framework based on graph variational autoencoder extracted embeddings from API-call graphs. *PeerJ Computer Science*, 8, e988. DOI: 10.7717/peerj-cs.988

Guo, Y. (2023). A review of Machine Learning-based zero-day attack detection: Challenges and future directions. *Computer Communications*, 198, 175-185. DOI: 10.1016/j.comcom.2022.11.001

Hussain, A., Saadia, A., Alhussein, M., Gul, A., & Aurangzeb, K. (2024). Enhancing ransomware defense: deep learning-based detection and family-wise classification of evolving threats. *PeerJ Computer Science*, 10, e2546. DOI: 10.7717/peerj-cs.2546

Krithika, E. (2024). A comprehensive literature review on ransomware detection using deep learning. *Cyber Security and Applications*, 100078. DOI: 10.1016/j.csa.2024.100078

Kumar, J., Rajendran, B., & Sudarsan, S. D. (2024). Zero-Day Malware Classification and Detection Using Machine Learning. *SN Computer Science*, 5, 93. DOI: 10.1007/s42979-023-02404-w

Kunwar, P., Aryal, K., Gupta, M., Abdelsalam, M., & Bertino, E. (2024). SoK: Leveraging Transformers for Malware Analysis. *arXiv preprint* arXiv:2405.17190. DOI: 10.48550/arXiv.2405.17190

Li, C., Lv, Q., Li, N., Wang, Y., Sun, D., & Qiao, Y. (2022). A novel deep framework for dynamic malware detection based on API sequence intrinsic features. *Computers & Security*, 116, 102686. DOI: 10.1016/j.cose.2022.102686

Liu, H., et al. (2024). MPDroid: A multimodal pre-training Android malware detection method with static and dynamic features. *Computers & Security*, 150, 104262.

Liu, J., Zhao, Y., Feng, Y., Hu, Y., & Ma, X. (2024). SeMalBERT: Semantic-based malware detection with bidirectional encoder representations from transformers. *Journal of Information Security and Applications*, 80, 103690. DOI: 10.1016/j.jisa.2023.103690

Maniriho, P., Mahmood, A. N., & Chowdhury, M. J. M. (2023). API-MalDetect: Automated malware detection framework for windows based on API calls and deep learning techniques. *Journal of Network and Computer Applications*, 218, 103704. DOI: 10.1016/j.jnca.2023.103704

Maniriho, P., Mahmood, A. N., & Chowdhury, M. J. M. (2024). A Survey of Recent Advances in Deep Learning Models for Detecting Malware in Desktop and Mobile Platforms. *ACM Computing Surveys*, 56(5), Article 117. DOI: 10.1145/3638240

Moreira, C. C. (2023). Improving ransomware detection based on portable executable header using Xception convolutional neural network. *Computers & Security*, 130, 103265. DOI: 10.1016/j.cose.2023.103265

Rahali, A., & Akhloufi, M. A. (2021). MalBERT: Malware Detection using Bidirectional Encoder Representations from Transformers. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. DOI: 10.1109/SMC52423.2021.9659287

Saracino, A., & Simoni, M. (2023). Graph-Based Android Malware Detection and Categorization through BERT Transformer. In *ARES 2023 Conference*, pp. 1-7. ACM. DOI: 10.1145/3600160.3605057

Umme, Z., Khan, A., & Rajarajan, M. (2022). Ransomware detection using deep learning based unsupervised feature extraction and a cost sensitive Pareto Ensemble classifier. *Scientific Reports*, 12, 15816. DOI: 10.1038/s41598-022-19443-7