Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MALWAREXAI-SECNET: AN EXPLAINABLE SEVERITY-AWARE MULTI-TASK DEEP LEARNING FRAMEWORK FOR INTELLIGENT MALWARE DETECTION AND SECURITY ANALYTICS

ANURADHA ANUMOLU[1*], ARUN SINGH CHOUHAN[2], S SRINIVASA RAO[3]

[1]Research Scholar, Department of Information Technology, Malla Reddy University, Hyderabad, Telangana India
[2]Associate Professor, Department of Computer Science and Engineering, Malla Reddy University, Hyderabad, India
[3]Professor and Principal, Muffakham Jah college of Engineering and Technology, Hyderabad, India

[*]Corresponding Author: anuradha.anu503@gmail.com

ABSTRACT

Modern malware is becoming increasingly obfuscated, polymorphic and behaved, posing challenges for reliable malware detection and security analysis. Current deep learning methods are mostly task-specific and lack the ability to jointly address the following problems in a single framework: malware detection, malware family classification, identification of security-related behavioral capabilities, assessment of the severity of the threat, uncertainty prediction, explanation and security decision support. To overcome these drawbacks, the authors have introduced a novel, explainable, severity-aware multi-task deep learning framework called MalwareXAI-SecNet for intelligent malware detection and security analytics.For this purpose, authors have presented an innovative explainable and severity-aware multi-task deep learning framework, dubbed MalwareXAI-SecNet, to overcome these drawbacks for intelligent malware detection and security analytics. The framework proposes a dataset-adaptive feature encoding that involves a gated residual multilayer perceptron (grid-MLP), a multi-scale one-dimensional convolutional neural network (mdCNN), and a transformer encoder, followed by adaptive gated-attention fusion. The missing label aware multi-task optimisation allows for learning from diverse datasets and reducing the conflicting predictions across the different tasks (detection, family classification, behavioral capabilities, and ordinal threat severity). The uncertainty estimation, probability calibration, and interpretable predictions are facilitated by Monte Carlo dropout, temperature scaling, SHAP, integrated gradients, and attention-based evidence. A detection F1-score of 99.46%, 98.69% on CIC-MalMem-2022 and EMBER2024, respectively, and family-classification Macro-F1 score of 97.31% on BIG 2015 are reported from experiments performed on CIC-MalMem-2022, EMBER2024 and BIG 2015, respectively. The framework also provides a family-classification Macro-F1 score of 98.51% and a behavioural-capability Macro-F1 score of 96.31% and a threat-severity Macro-F1 score of 96.72% on CIC-MalMem-2022. The results show the potential of the proposed framework in providing a comprehensive malware intelligence from predictive performance, behavioural interpretation, severity-based risk prioritisation and confidence-aware security recommendations. The framework can help security analysts prioritise suspicious files and direct files in doubt or high-risk cases for further analysis.

Keywords: *Malware Detection, Multi-Task Deep Learning, Explainable Artificial Intelligence, Threat Severity Prediction, Security Analytics*

1. INTRODUCTION

Malicious files continue to evolve, with ever-changing behaviour, packing, polymorphism, and increasingly complex methods of execution making reliable detection and security analytics a cybersecurity challenge. Unfortunately, conventional signature-based systems are ineffective against new threats they have not been exposed to, or against rapidly evolving threats; many machine-learning approaches still rely on handcrafted features or simple single-task concerns. Lightweight Transformers, CNN-based hierarchical classifiers, hybrid CNN–LSTM architectures, explainable ensembles, few-shot learning, severity-aware detection and graph neural networks have consequently been

investigated [1]–[7]. While these studies have shown that deep learning can significantly enhance malware recognition, most solutions continue to focus on individual aims (e.g., binary detection, family classification, explainability) and do not offer an all-in-one security/intelligence pipeline.

The literature also identifies several key gaps. While lightweight Transformer approaches improve contextual modelling, they generally only work with detection and local explanation [1]. However, their hierarchical CNN hybrids are limited to closed-set classification and are useful for detection and family-level attribution [2]. There are detection approaches that consider severity, as in MADESANT, which adds a risk assessment module. However, these do not all consider and integrate the various features of malware family, behavioural capability, calibrated uncertainty and cross-task consistency at the same time [26]. Multimodal approaches like GIT-GuardNet are useful by combining static, graph, and temporal information, and explainable methods like ANAKIN and robust ensemble systems boost the interpretability and resilience [33,35,40]. However, the lack of a consolidated unified framework that can collectively learn malware detection, family classification, malicious capabilities, ordinal severity, uncertainty, explanation, and response-oriented risk prioritisation remains poorly addressed.

To overcome these weaknesses, this work proposes a severity-aware multi-task deep learning framework, MalwareXAI-SecNet, for intelligent malware detection and security analytics with explainability. It aims to be flexible enough to accommodate heterogeneous representations of malware but also powerful enough to enable detection by binary, automated classification by malware family, identification of malware capabilities, and prediction of threat severity by malware order. New aspects are the introduction of dataset-adaptive encoding via gated residual MLP, multi-scale 1D-CNN, and Transformer branches, as well as the adaptive gated-attention fusion, the incorporation of a missing-label-aware multi-task optimisation, the use of behaviour-grounded severity annotation, cross-task consistency regularisation, Monte Carlo dropout-based uncertainty estimation, and probability calibration, along with explainability analyses that are tailored to representations by utilising SHAP, integrated gradients or attention evidence, as well as interpretable confidence-aware risk prioritisation.

MalwareXAI-SecNet has fourfold contributions in the field of research. First, it proposes a design to adaptfully map heterogeneous malware representations, such as structured memory attributes, executable metadata, byte sequences, and assembly/opcode sequences, into a common latent interface while maintaining the dataset-specific preprocessing and encoding. What is not just being used is multiple encoders, but the exploration of whether representation specific encoding and sample adaptive fusion could enable a shared downstream analysis pipeline while keeping incompatible raw feature spaces distinct.

Second, it proposes a missing label-aware multi task optimisation for malware detection, family classification, behavioural-capability identification and ordinal severity prediction. Each task is used for learning only when its ground-truth labelling is available, solving a problem of incomplete and non-uniform ground-truth supervision of the selected datasets.

Third, it uses explanations, probability calibration, uncertainty estimation, and cross-task consistency regularisation. This provides a basis for questioning not only the validity of individual predictions but also the validity of related outputs and the confidence they should be relied upon to inform risk prioritisation.

Fourth, it examines the framework from the perspective of predictive measures and across a variety of other analyses of ablation, calibration, robustness, explanation behaviour, and computational costs. The desired knowledge contribution is empirical, that is, the goal is to understand how these mechanisms work together when implemented, and what elements are important for their predictive performance and decision support. The expression of individual components' benefits is made only up to the verified experiments.

The rest of this paper is structured like this. The literature on the subject is reviewed in Section 2, where the major gaps in the research are mentioned. The proposed methodology is introduced in Section 3, alongside the datasets, adaptive encoders, fusion strategy, multi-task learning, severity modelling, uncertainty-aware explainability, algorithms and experimental protocol. The full experimental results, including

predictive, robustness, calibration, explainability, risk, scalability, and statistical analysis, are presented in Section 4. The results and the limitations of the study are presented in Section 5. The work is concluded in Section 6 and outlines future research directions.

2. RELATED WORK

The field of malware detection research has shifted from traditional machine learning classifiers to representation learning models that can capture spatial, sequential, behavioural and contextual properties. Lightweight Transformers have enabled the modelling of long-range dependencies with local explanations, and Hierarchical CNN hybrids have enabled the move from binary detection to family and subfamily attribution of malware [1] [2]. The issues of imbalance and error propagation from detection to family classification have also been tackled by regularised hierarchical learning [3]. More extensive studies related to LLM-based malware analysis stress the importance of contextual reasoning and security-focused fine-tuning, but leave robustness, computational efficiency, and standardised explanations unaddressed [4]. For malware (image) classification, parallel branches based on CNN with channel attention and LSTM modelling have been used, and PCANNs+LSTM have shown good results, which proves the complementarity of spatial and sequential representations [5]. It is also known that reported performance is highly specific to the type of feature investigated, the distribution of the families, the extent of missing behaviours and the composition of the datasets investigated – these are all types of DataSet-centric investigations.

New learning methods are increasingly focusing on limited labels and moving malware. Few-shot prototypical learning with feature selection, drift detection, and XAI has been proven to perform well with few training samples [7], and visualisation-based approaches offer other representations of the malware structure but are susceptible to obfuscation and adversarial modification [8]. Self-supervised learning diminishes reliance on costly malware labelling by using contrastive representation learning [9]. Evaluations of static, dynamic, and hybrid deep learning techniques also show the need for hybridising complementary representations for zero-day and evolving threats [10]. Recent attempts to enhance detection while managing dimensionality involve behaviour-based machine learning [11], correlation-driven feature selection (for LSTM) [12], entropy-oriented feature selection [13], and deep-belief architectures that merge signature and behavioural information [14]. System metrics and network logs can also be used for visualisation and decision support for security purposes [15], and dynamic deep learning can be used along with heuristics to improve malware detection and family analysis compared with an isolated static approach [16]. However, consistent challenges are outlined in the comprehensive evidence, including adversarial robustness, computational complexity, limited labelled data, and interpretability [17].

Dynamic and adaptive strategies have therefore been widely studied. Deep network analysis has been explored for detecting behavioural malware [18], and GANs generate representations for training the network on previously unknown malware [19]. The problem of limited malware samples and class imbalance is addressed by Siamese one-shot learning [20]. At the same time, detection can be further improved by combining it with deep-forest features and ensemble learning to detect Android malware [21]. Smart-home studies focus on early warning and intrusion detection, but they also adopt ensemble mechanisms; in cyber-threat intelligence frameworks, they move toward classification for IoT security monitoring. PCA, API information, clustering and anomaly detection are still efficient conventional Android classifiers [24]. In addition, BERT-based hybridisation has proved its applicability in IoT malware traffic [25].

One key shift, in particular, is from malware identification to security intelligence. Severity-aware analysis has been introduced through cascading neural-based detection and risk factor assessment [26], and double-tier AI models have been introduced for malware and ransomware detection in healthcare-oriented SDN environments [27]. Memory-efficient models such as lightweight ANNs, SVMs, and boosting are highlighted, while BERT ensembles incorporate permission features for advanced Android threat detection [29]. Dynamic-feature SVR shows the benefits of nonlinear behavioural modelling [30], and fuzzy ensemble systems fuse classifier outputs using fuzzification-based decision-making [31]. Permission-to-exploitation

mapping also enhances scalability and adversarial resilience, though permission-only representations might not include more elaborate malware behaviours [32].

Multimodal fusion and interpretability are trending areas of research these days. Transformers with static code, call graphs, and temporal traces come in the form of Graph-informed Transformers, and they show effectiveness in the presence of obfuscated and stealthy malware [33]. Local visual signatures are combined with global contextual modelling using hybrid CNN–Transformer and Grad-CAM for interpretation [34]. Graph-based Malware Detection uses API-call graphs and GNNExplainer for revealing significant APIs and control-flow relationships [35]. Explainable stacking ensembles use SHAP and LIME to enhance the transparency of industrial malware classification [36], and CNN–BiLSTM–self-attention hybridisation improves representation learning of evasive Android malware [37].

However, explainability is a key mandate for a good and reliable malware analysis [38]. A detailed investigation of the system's importance with SHAP- and LIME-based methods reveals that such methods are beneficial for analysts' understanding but also highlights the need to consider the robustness and reliability of the explanation in use [39]. In a recent ensemble system, static malware detection, adversarial defence, XAI, and deployment feasibility are explored together [40]. Collectively, the literature shows strong progress in detection, attribution, multi-modal representation, severity assessment, robustness, and explainability; but these capabilities are largely disjoint. SecNet does not leverage explainability of multi-representations, nor does it involve uncertainty calibration, nor does it involve risk prioritisation, nor does it use confidence to inform security decisions; these are all features that have not been well studied and that MalwareXAI-SecNet investigates. The importance of explainability, interpretability, and trustworthy decision-making has been a growing priority in recent studies involving the use of AI in cybersecurity. In security-sensitive environments, the reliability and robustness of explanations in adversarial XAI are crucial as they are essential for security.In the context of Industry 5.0, Khan et al. [44] conducted a systematic review of XAI-based intrusion detection systems (IDSs), emphasizing the significance of explanation reliability and robustness in security-sensitive scenarios. Likewise, Ofusori et al [45] explored explainability and interpretability in cybersecurity applications, underscoring the importance of transparent AI decisions that security analysts can comprehend and assess. Nalinipriya et al. [46] explored the use of explainable AI for early detection and mitigation of cyber threats in large-scale network settings, highlighting the importance of incorporating interpretable predictions into real-world security practices. Recent research by Mystakidis et al. [47] also showed the importance of explainable learning in memory-based malware analysis, with their proposed framework being alert-driven framework for malware detection based on memory artefacts, TabNet and ensemble classification. In total, these studies highlight a shift toward interpretable cybersecurity models, explainable threat detection and analyst oriented decision support. But there are still some important directions to explore on how to integrate these five aspects of malware detection, family classification, behavioural-capability identification, ordinal threat-severity prediction, missing-label-aware multi-task optimisation, cross-task consistency regularisation, and uncertainty-aware risk prioritisation. Inspired by this gap, MalwareXAI-SecNet gathers heterogeneous feature encoders, adaptive gated-attention fusion, multi-task prediction, calibrated uncertainty estimation, representation-specific explanations in a single framework of malware security analytics.

Problem Statement and Research Gap

Despite the progress of deep learning, multimodal feature extraction, and explainable artificial intelligence, malware research is still struggling to combine extensive security intelligence with a predictive performance. Most of the current methods are heavily focused on specific tasks like binary malware detection, family classification, behavioural analysis or explanation generation. While recent work has made significant progress in explainable intrusion detection [44]–[45] or cybersecurity interpretability [46]–[47] or large-scale cyber-threat mitigation, none of these advances provides a comprehensive solution to the combined problem of malware detection, malware family classification, identification of behavioural capabilities, prediction of ordinal threat severity, uncertainty estimation, and confidence-aware security decision support. In

addition, heterogeneous malware datasets come with varying feature representations and only partial task annotations which makes joint learning challenging without introducing incompatible feature space or unreliable supervision. There can also be conflicting results across independent prediction models, and uncalibrated levels of confidence and poorly validated explanations may constrain the confidence of downstream security decisions.

The research problem in this study is therefore centered on how to design a single, dataset-adaptive learning framework that would be able to use the disparate representations of malware, to learn multiple related security tasks from partially available labels, to minimize label inconsistencies between task predictions, and to produce interpretable and uncertainty-aware threat assessments. Solving this challenge goes beyond detection accuracy and involves creating meaningful links between the types of malware, what it can do, how dangerous it is, how well it can predict and how it contributes to operational risk prioritisation.

In this study, the research gap solved is the lack of joint evaluation of heterogeneous malware representations, incompleteness of task labels, lack of consistency of cross-task predictions, and confidence-aware security interpretation. MalwareXAI-SecNet considers the following four questions: (i) can dataset-adaptive encoders and sample-specific gated-attention fusion also provide a shared learning interface while preserving the semantics of different feature representations; (ii) for related detection, family, behavioural-capability, and ordinal severity tasks, can missing-label-aware optimisation also support these tasks without assuming unavailable labels as ground truth; (iii) can consistency regularisation reduce incompatible predictions across the aforementioned tasks; and (iv) how calibration, uncertainty estimation, and representation-specific explanations can be applied in conjunction with predictive outputs for risk prioritisation.

The new knowledge that is desired, therefore, is not for the assertion that each individual technique is new by itself, but for a scientifically valid knowledge of them as part of a multi-task malware analysis workflow. The contribution should be evaluated based on the task-level performance, ablation results, calibration and robustness analysis, explanation assessment, and computational measurements. The datasets are not necessarily represented in the same way and have different label spaces; conclusions regarding cross-dataset generalisation then have to be restricted to the experiments actually conducted and cannot infer the superiority of the framework generally or that all tasks are jointly supervised across all datasets.

Table 1. Literature comparison and research gaps addressed by MalwareXAI-SecNet

Representative Study / Approach	**Main Focus**	**Limitation Relevant to This Study**	**Gap Addressed by MalwareXAI-SecNet**
Lightweight Transformer-based malware analysis [1]	Contextual malware detection and local explanation	Focuses on narrower prediction and explanation objectives.	Integrates multiple malware-analysis tasks with reliability assessment.
Hierarchical CNN-based classification [2]	Malware detection and family-level attribution	Does not establish a combined treatment of severity, uncertainty, and cross-task consistency.	Combines family classification with behavioural-capability and ordinal severity prediction.
Severity-aware detection, such as MADESANT [26]	Malware detection with risk assessment	Does not cover all the proposed tasks and reliability mechanisms together, according to the reviewed literature.	Investigates severity-aware multi-task learning and consistency between related predictions.
Explainable malware	Interpretation of model	Explanation quality and	Combines represent

analysis [38]–[39]	predictions for security analysis	predictive performance alone do not establish calibrated confidence or consistency across multiple tasks.	ation-specific explanations with uncertainty estimation and probability calibration.
Recent explainable cybersecurity and malware studies [44]–[47]	Explainability, threat mitigation, and alert-driven malware analysis	These studies address related but differing security problems; their methods and evaluation settings are not directly interchangeable.	Evaluates an integrated malware-analysis workflow involving multiple prediction tasks and risk prioritisation.

To highlight the research gap tackled by this study, Table 1 compares representative approaches based on their main objectives, limitations related to the current research and the gap studied by MalwareXAI-SecNet. The comparison illustrates that malware detection, family attribution, and severity assessment have been addressed in previous studies, as has been explainability, with goals and evaluation contexts differing. The motivation for the proposed framework is not that these individual techniques are not studied but rather the use of multiple techniques for multi-task malware analysis, cross-task consistency, calibrated uncertainty, and risk-oriented decision support gives reason to evaluate them in a systematic fashion. This comparison provides the justification for the framework proposed and differentiates the existing capabilities of previous approaches from the specific integration examined in this study.

3. MATERIALS AND METHODS

The detailed methodology of the MalwareXAI-SecNet framework for intelligent Malware detection and security Analytics is presented. It algorithmically discusses the publicly available datasets used, approaches to preprocessing them, multiple views of input features, dataset-dependent deep feature encoders, adaptable fusion, multitask learning, severity modelling, uncertainty estimation, explainability, risk prioritisation, security decision support, algorithmic workflow, and experimental evaluation protocol.

In this section (3.1), the overall MalwareXAI-SecNet architecture is presented, and the problem is formulated.

Malware detection can be cast as a supervised security analytics problem where the question is: Is this malware or not; if it is, what kind of malware is it; if it's that kind of malware, what payload or modification does it carry; and if it does, how does it operate? Let the samples for the input be x_i, and its dataset-specific feature space be denoted by $\mathcal{X}$. Based on the source of data sets x_i can include memory-forensic data, executable metadata, or sequences of bytes/assembly/opcode sequences or related malware descriptions. The complete target for sample i in Eq. (1).

$$\mathcal{Y}_i = \{ y_i^{(d)}, y_i^{(f)}, y_i^{(c)}, y_i^{(s)} \} \quad (1)$$

In Equation (1), $y_i^{(d)}$ represents the binary malware-detection label, $y_i^{(f)}$ the malware-family label, $y_i^{(c)}$ a multi-label behavioural-capability vector and $y_i^{(s)}$ the ordinal threat-severity label. Superscripts $(d), (f), (c), (s)$ refer to detection, family, capability and severity tasks.

The common representation z_i is used to map to the probability that the i -th sample is malicious in the context of malware detection in in Eq. (2).

$$\hat{y}_i^{(d)} = \sigma(W_d z_i + b_d) \quad (2)$$

The trainable weight matrix W_d and bias vector b_d of the detection head, sigmoid actiation function $\sigma(\cdot)$ and predicted malicious probability $\hat{y}_i^{(d)}$ are here. Malware-family classification is the process of placing a malware sample into one of K_f malware families defined in the malware taxonomy. The family-probability vector is

$$\hat{y}_i^{(f)} = softmax(W_f z_i + b_f), \quad (3)$$

Additionally, W_f and b_f in Equation (3) are trainable parameters of the family head, and $\hat{y}_i^{(f)}$ a probability vector of length K_f, whose biggest component represents the predicted family.

Behavioural-capability identification can be realized as a multi-label prediction problem, as multiple behaviors may be observed in a single sample, such as persistence, code injection, privilege escalation, credential access, data exfiltration, command-and-control communication, defense evasion, destructive activity, and so on. The number of behavioural capabilities is denoted by K_c, and the capability-probability vector is in Eq. (4).

$$\hat{y}_i^{(c)} = \sigma(W_c z_i + b_c), \quad \hat{y}_i^{(c)} \in [0,1]^{K_c}, \quad (4)$$

In this case W_c and b_c are capability-head parameters, and each capability value $\hat{y}_i^{(c)}$ is an independent probability of the presence/absence of one of the K_c malicious capabilities.

The severity of threats is represented as an ordinal task instead of independent classes so as to maintain the relation among the severity levels. A severity level could be considered as in Eq. (5).

$$y_i^{(s)} \in \{0,1,2,3\}\{Low, Moderate, High, Critical\}. \quad (5)$$

The numbers 0, 1, 2 and 3 represent Low, Moderate, High and Critical. The ordered coding allows prediction errors to be reflective of the distance between severity levels and lends itself to the use of the ordinal-learning formulation described later (Section 3.4).

MalwareXAI-SecNet is self-contained and dataset-adaptive. Two encoders are used; encoders are selected based on the structure of the input representations. Let $E_\theta^{(r)}$ be the encoder corresponding to r being the representation type r and θ being the trainable parameters of the encoder. The view specific latent representation is:

$$h_i^{(r)} = E_\theta^{(r)}(x_i) \quad (6)$$

As in Equation (6), $h_i^{(r)}$ is the latent feature vector that is produced by encoder of type r for sample i. The gated residual MLP is used for structured memory/executable attributes, while the multi-scale 1D-CNN will be responsible for byte sequences, and the Transformer encoder for sequences of behaviours or sequences of assembly/opcodes. There is no need to come force incompatible raw features to a single preprocessing space.

Let $h_i^{(1)}, \ldots, h_i^{(V)}$ be the V different view-specific latent representations, when the sample has multiple feature views. An importance score $g(\cdot)$ is given to each view in the trainable scoring function. The attention-coefficient for view v normalized is in Eq. (7).

$$\alpha_i^{(v)} = \frac{exp(g(h_i^{(v)}))}{\sum_{j=1}^{V} exp(g(h_i^{(j)}))} \quad (7)$$

Here, $\alpha_i^{(v)}$ is the normalized contribution of view v for sample i and the denominator normalizes the scores over all V available views. The weighted aggregation is then used to obtain the shared malware representation.

$$z_i = \sum_{v=1}^{V} \alpha_i^{(v)} h_i^{(v)} \quad (8)$$

z_i corresponds to the common embedding of malware given to all four prediction heads in Equation (8). Unlike fixed fusion weights, as the weights $\alpha_i^{(v)}$ are sample specific, the framework can highlight the contribution of evidence sources differently for different malware cases.

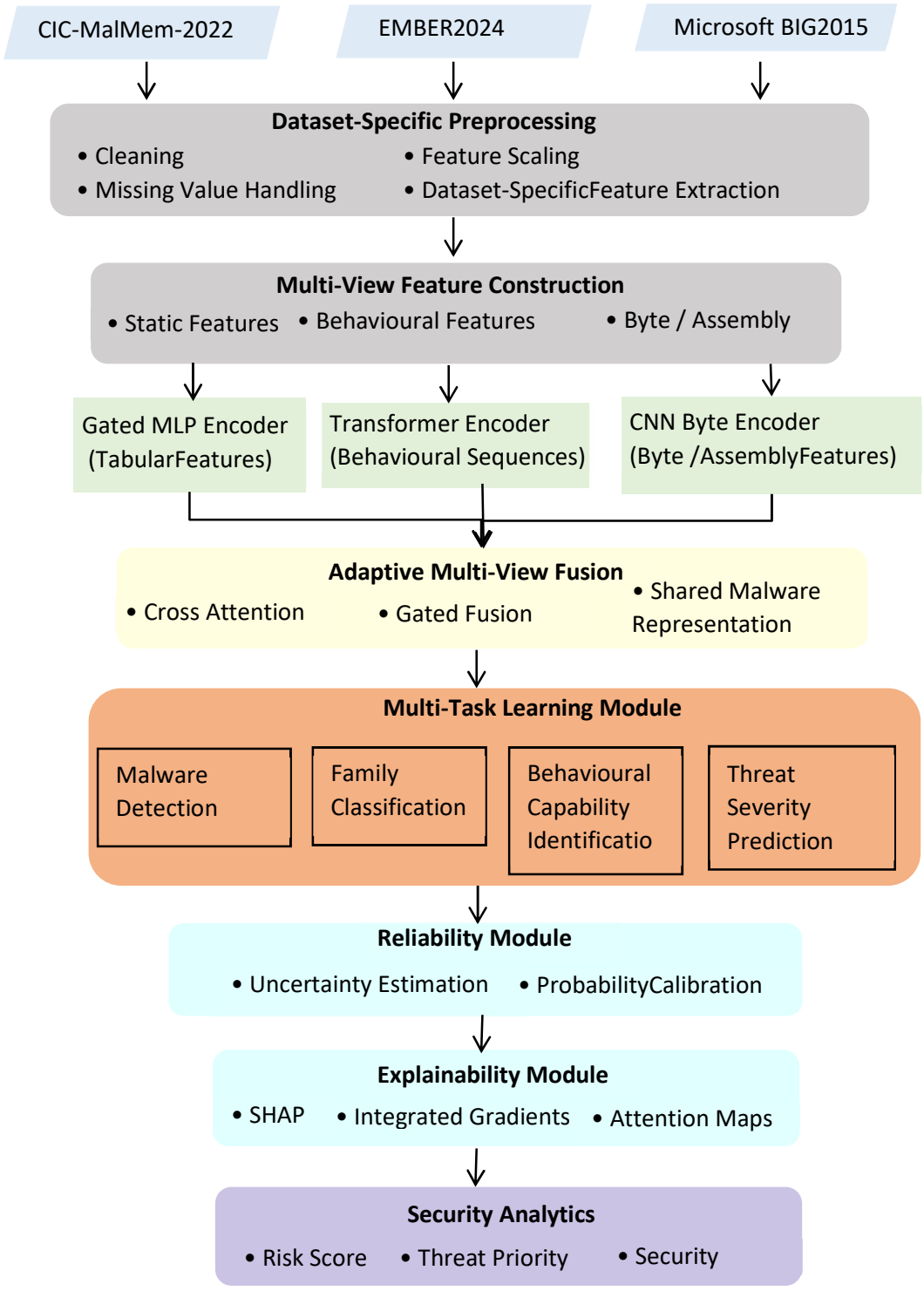


Figure 1: Overall Architecture Of Malwarexai-Secnet For Explainable Multi-Task Malware Detection And Security Analytics

Thus, the complete pipeline goes from dataset-specific pre-processing, multi-view feature construction, to dataset-adaptive encoding, attention-based fusion and the creation of the shared embedding z_i. All the heads of detection, family and capability and severity are motivated by this representation. These outputs are then used in discovering missing labels in multi-task

optimization, in cross-task consistency regularization, in uncertainty estimation and calibration, explainability, risk prioritization and security decision support. The entire system architecture of MalwareXAI-SecNet is shown in Figure 1, along with the data flow among these components.

3.1 Datasets and Data Preprocessing

The experimental test of MalwareXAI-SecNet uses three public malware datasets that complement each other: CIC-MalMem-2022 [41], EMBER2024 [42] and Microsoft Malware Classification Challenge BIG 2015 [43]. These datasets were chosen because together, they can give memory-forensic attributes, features and metadata of executables, malware family, behavioural labels, hexadecimal formatting of bytes, and assembly level information. What is more important is that none of these three datasets are directly merged as there is a significant difference between the platform coverage, representation of input and dimensionality of features, label space, and collection methodology. Instead, every dataset goes through their respective preprocessing and dataset adaptive encoding directions, and the common downstream MalwareXAI-SecNet architecture offers a same framework for representation learning, multi-task prediction, explainability, and security analytics. Table 2 presents the collected data, describing what they are and their usage in the study.

CIC-MalMem-2022 [41] is used as a key experimental dataset as it offers an equilibrated ensemble of memory forensic observations for both benign and malicious executions. There are 58,596 thoughts in the dataset, with an equal number of benign (29298) and malicious thoughts (29298). It consists of 15 types of malwares in terms of three major categories: Trojan Horse, Spyware and Ransomware, and five families within each category. All of these make CIC-MalMem-2022 especially well-suited for binary malware detection, discrimination at the category level, classification at family level, explainability, and generation of behaviour-driven severity labels. The preprocessing pipeline initially filters out redundant or uninformative observations, and analyzes the attributes to check for missing or invalid values in each one. If numerical values are missing, then they are estimated only from the training partition to prevent information leakage. Robust normalization is applied for continuous features; categorical or discrete ones are encoded as needed. Next, using related memory-forensic features, semantically meaningful groups of processes, memory regions, handles, modules, services, injected activity, and runtime artefacts are formed. The multiple input views are generated by these groups for the proposed feature encoder.

The large-scale external validation data is EMBER2024 [42]. It includes hashes, labels, metadata and feature representations of the over 3.2 million malicious and benign files across six file formats. The dataset was specifically tailored for multiple malware-classification use cases such as malware detection, malaria-family classification and malware-behaviour identification. It also includes a challenge set of malicious files which were previously undetected by antivirus software to evaluate the product under more challenging conditions when evasive malwares are involved. In MalwareXAI-SecNet, only the static features of the Executable are passed that include executable Characteristics, Structured Data, Metadata, Strings, Imports, Exports and Task specific labels based on the chosen prediction objective. Only the training part is used to estimate the feature normalization parameters. No arbitrary random split is imposed on the temporal structure of the dataset; in this way realistic temporal distribution changes can be evaluated. The evasive-malware challenge samples are not used in the normal model fitting but are kept aside to assess the model's robustness.

Microsoft Malware Classification Challenge BIG 2015 [43] offers variant and distinct representation of malicious software which is used as an independent benchmark for malware-family classification. Over 20,000 malware samples are included in the original challenge data set as hexadecimal representations of the malware bytecode or, in the case of disassembled binaries, disassembled assembly files and is made available exclusively for malware-family classification. In its labeled training set, it incorporates 9 malware families, which is especially useful to analyze and evaluate whether the introduced representation-learning architecture is able to separate out the fine-grained characteristics of each family. Hexadecimal files are processed to become lists of ordered sequences of bytes, which are free of address markers and the non-printable tokens. If

longer sequences can't fit within that maximum length set as the model configuration, they are split into smaller parts or they are simply cut; if a sequence is smaller than that maximum length, they are padded with a dedicated padding token. The assembly files are separately processed to get the instruction and opcodes sequences along with structures etc. The output representations are then fed to the multi-scale 1D-CNN and Transformer based encoding pathway, respectively. Because there is no matched benign class in BIG 2015 it can only be used for family-classification experiments, and will not be used to support the binary-malware detection task in Table 2.

Table 2: Summary Of Public Datasets Used For The Development And Evaluation Of Malwarexai-Secnet

Dataset	Data representation	Labels/classes	Role in the proposed study	Dataset URL
CIC-MalMem-2022 [41]	Memory-forensic tabular features	Benign/malicious; Trojan, Spyware and Ransomware categories; 15 malware families	Primary dataset for malware detection, category/family classification, explainability, severity modelling, and risk analytics	https://www.unb.ca/cic/datasets/malmem-2022.html
EMBER2024 [42]	Static executable features, metadata, hashes, and multi-format file representations	Malware detection, family, behaviour and other task-specific labels	Large-scale external validation, behavioural-capability analysis, temporal generalization, uncertainty evaluation, and evasive-malware robustness	https://github.com/FutureComputing4AI/EMBER2024
Microsoft BIG 2015 [43]	Hexadecimal byte files and disassembled assembly files	Nine malware families	Independent malware-family classification, byte/assembly representation learning, and sequence-level evaluation	https://www.kaggle.com/c/malware-classification/data

Thus, the preprocessing approach is intentionally designed to not create a pseudo-common feature space of the raw data sets. CIC-MalMem-2022 offers memory-behaviour evidence, EMBER2024 offers large scale executable/behavioural validation, and BIG 2015 offers byte and assembly level family discrimination. The representations of these datasets are then adaptively mapped to a common dimensionality, also as described in Section 3.3, whilst still keeping MalwareXAI-SecNet a unified methodological design without actually merging incompatible observations.

3.2 Multi-View Feature Representation and Dataset-Adaptive Deep Encoding

To avoid heterogenous datasets being forced into a single raw feature space, MalwareXAI-SecNet renders each input sample as a collection of multiple complementary feature views to cover the structural, behaiourial, sequential and low-level code characteristics. Let the i-th malware sample be represented as a set of V views in Eq. (9).

$$\mathcal{X}_i = \{ x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(V)} \} \quad (9)$$

where $x_i^{(V)}$ are semantically different representations. CIC-MalMem-2022 mostly

bases its views on behavioural and memory-forensic features like process activity, handles, loaded modules, services, indicators of injected-code, and other runtime features. The views for EMBER2024 contain static (executable) properties, structural metadata, imported and exported symbols, strings, section-level info and other characteristics of the file. Microsoft BIG 2015 provides two code-oriented views: hexadecimal byte sequences, and the disassembled sequence of assembly/opcode code. This decomposition at the view level, ensures that each of the data sets preserves their native semantics and also provides complementary malware-related information to the downstream encoder.

Static-feature view are structured vectors that include the executable static features' attributes as well as the metadata's. Behavioural and memory views include features that mirror activity at run time or at the memory level and are structured by their functional groups, instead of lumped together into an undifferentiated vector. Recurrent motifs of bytes, headers, packed regions, local signatures, etc. can be learned, directly from the sequence because the local ordering of hexadecimal tokens is preserved in the byte representations. After discarding irrelevant address details and other non-semantic artefacts, the order of instructions or opcodes in the assembly representations is retained. These sequences are encoded into token indices, and then deep encoded into a compact vector space. The first embedded representation for a sequence of T tokens are in Eq. (10).

$$E_i = [\, e_{i,1}, e_{i,2}, \dots, e_{i,T} \,], \quad e_{i,t} \in \mathbb{R}^{d_e} \qquad (10)$$

where d_e is the size of the token embedding dimension.

The gated residual multilayer perceptron processing structured/tabulary malware features. Before nonlinear transformation, the learning process in the gating mechanism filters out noisy or not so informative inputs, while effectively highlighting the utility of each feature relative to the others. The feature gate for an input feature vector x is given by in Eq. (11) and (12).

$$g = \sigma(W_g x + b_g) \qquad (11)$$

and the gated representation is derived as:

$$\tilde{x} = g \odot x \qquad (12)$$

Where is the element wise multiplication with $\odot$. The gated representation then goes through the residual nonlinear layers to enable the learning of feature interactions at higher levels without discarding any information. The residual transformation at layer l is given as in Eq. (13).

$$h^{(l+1)} = h^{(l)} + \varphi(W^{(l)} h^{(l)} + b^{(l)}) \qquad (13)$$

where, the nonlinear activation function is denoted as $\varphi(\cdot)$. The most informative latent attributes are then highlighted by 'feature-wise attention'. Such a mix of gating, residual learning and attention is designed to enhance the models' ability to resist quantities of redundant or weakly discriminative or correlated tabular features.

The multi-scale one-dimensional convolutional encoder is used to process malware representations of only a small set of bytes. Several different convolutional kernels with different receptive fields are applied in parallel, allowing short, medium, and longer-range local byte patterns to be found at the same time. The local representation of the k-th convolutional branch centered at position t is a scalar quantity calculated as the width r_k of the k-th branch of the convolutional model is equal to the bandwidth of the value k function in Eq. (14).

$$c_{i,t}^{(k)} = \varphi(W_c^{(k)} * E_{i,t:t+r_k-1} + b_c^{(k)}) \qquad (14)$$

This is using the one-dimensional convolution operation indicated by the star. All the output of the receptive fields are aggregated and connected to create a compact representation of the malware, at the level of bytes. The flexibility to use multiple receptive field sizes allows the encoder to capture not only the code signatures that are more localized in nature, but also the more distant neighbouring constructions of bytes without having to fix a kernel scale parameter.

The Transformer encoder is used to model sequential behaviour information and assembly/opcode representations. Self-attention is not guaranteed to retain the order of the sequence, as such positional information is provided as inputs to the token embeddings prior to attention processing. The position aware input is represented as in Eq. (15).

$$Z_{i,t}^{(0)} = e_{i,t} + p_t, \qquad (15)$$

where p_t is the positional encoding at position t. Each attention head outputs its respective attention, query, and value representations via a

trainable linear transformation of $Z^{(0)}$. The scaled self-attention operation is given by in Eq. (16).

$$Attn(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \quad (16)$$

where d_k is the dimension of the key representation. The encoder has multiple attention heads which process input in parallel so the model can learn various types of long-range dependency between the behavioural events, instructions, or opcode patterns. The resulting contextual representation is then fed through feed-forward layers and residual normalization, to form the final embedding for the sequence.

The three sets of data have outputs of the encoders and statistics of different dimensionalities, and a projection adapter corresponding to each dataset is therefore proposed prior to adaptive fusion. The outputs $h_i^{(r)}$ from each encoder are transferred to a common latent space of dimension d_z, by in Eq. (17).

$$u_i^{(r)} = \varphi(W_p^{(r)} h_i^{(r)} + b_p^{(r)}), \quad u_i^{(r)} \in \mathbb{R}^{d_z}, \quad (17)$$

Dataset- or representation-specific projection parameters, $W_p^{(r)}$ and $b_p^{(r)}$. This adapter exposes a single latent interface, with the representation-specific paths for encoding. A common feature space can then be used to express structured memory features, executable metadata, byte-level patterns as well as assembly/opcode sequences, without actually merging incompatible raw attributes. The representations as projected are then fed to the adaptive multi-view fusion mechanism, mentioned in Section 3.4.

3.3 Adaptive Multi-View Fusion and Multi-Task Malware Learning

An adaptive gated-attention fusion scheme is used to fuse the projected view-specific representations derived from Equation (17). This stage is for estimating the significance of each available view for each individual sample, discarding irrelevant or noisy views and combining information from related views into a single malware embedding. This mechanism is different from plain old concatenation or fixed weight fusion, which gives the model the opportunity to use different features sources with different weights for different cases. The attention relevance score for the v-th projected representation $u_i^{(v)}$ is calculated as in Eq. (18).

$$s_i^{(v)} = q^T \tanh(W_a u_i^{(v)} + b_a), \quad (18)$$

where the W_a and b_a are can be trained attention parameters, and q is a learnable context vector. The relevance scores are normalized over the V available views and sample specific attention coefficients are obtained, in Eq. (19).

$$\alpha_i^{(v)} = \frac{exp(s_i^{(v)})}{\sum_{j=1}^{V} exp(s_i^{(j)})}, \quad (19)$$

Here, $\alpha_i^{(v)}$ is the relative contribution of the view v for the sample i. Concurrently, a feature-wise gate for each envisioned representation is learned, allowing to suppress "non-sense" or "uninformative" dimensions for the set of samples currently being taken into account. The gating vector is the following in Eq. (20).

$$g_i^{(v)} = \sigma(W_g u_i^{(v)} + b_g), \quad (20)$$

The attention coefficient regulates the importances of the views on level of attention while the gating vector regulates the features-level within each view. The final unified malware embedding is then a weighted and aggregated version of the gated representations as in Eq. (21).

$$z_i = \sum_{v=1}^{V} \alpha_i^{(v)} (g_i^{(v)} \odot u_i^{(v)}), \quad (21)$$

where $\odot$ means multiplication element-wise. Embedding in equation (21) can generate the common malware embedding method and decreases the influence of weak or redundant views at the same time preserving complementary information. The embedding serves as shared input to the four prediction heads optimized for each task. The malicious-benign head (the malware-detection head) classifies malware as malicious or benign, the malware-class head (the multiclass-family head) classifies malware into a set of various classes (malware families), the malware-behaviour head (the multi-label-head) classifies malware into a set of various behaviours, and the malware-ordered-severity head (the ordinal-severity head) estimates the ordered severity (Low, Moderate, High, Critical). They predict their maps with no duplication of the task formulations introduced in Section 3.1, using their own mappings, which are described below.

Due to the different sets of labels in the three experiments, MalwareXAI-SecNet adopted multi-task-learning with missing labels. A binary label availability mask is added for task t, sample i in Eq. (22).

$$m_i^{(t)} \in \{0,1\}, \quad (22)$$

The number of 1 in the mask shows an available ground-truth label, the number of 0s shows an unavailable ground-truth label. The mask therefore helps to stop unavailable target from being part of the optimization. Define $\ell_i^{(t)}$ as the loss of the i-th sample of task t. Normalized Loss: Loss of each task is computed only on the samples where annotations are valid in Eq. (23).

$$L^{(t)} = \frac{\sum_{i=1}^{N} m_i^{(t)} \ell_i^{(t)}}{\sum_{i=1}^{N} m_i^{(t)} + \varepsilon}, \quad (23)$$

Where ε is a small positive constant in order to avoid division by zero. The loss functions are chosen based on the type of prediction. For detection of malware, binary cross-entropy is used, for classification of malware family, categorical cross-entropy is employed and multi-label binary cross-entropy for prediction of behavioural capability is used. When learning about severity, the four ordered levels are converted to cumulative threshold targets (ordinal severity learning). The ordinal target is specified as follows for a threshold k in Eq. (24).

$$r_{i,k}^{(s)} = I(y_i^{(s)} > k), \quad k = 0,1,\dots,K_s - 2, \quad (24)$$

with the indicator function $I(\cdot)$ and K_s a number of ordered severity levels. The severity head estimates the probability that for each threshold, the true severity will exceed it, and the ordinal loss is calculated by calculating the binary cross-entropy of the severity over the moving targets in Eq. (25).

$$L^{(s)} = -\frac{1}{N(K_s-1)} \sum_{i=1}^{N} \sum_{k=0}^{K_s-2} m_i^{(s)} [r_{i,k}^{(s)} \log p_{i,k}^{(s)} + (1 - r_{i,k}^{(s)}) \log(1 - p_{i,k}^{(s)})]. \quad (25)$$

Adaptive task-loss weighting is used to avoid having one task with more numerals losing weight due either to sample availability or to large loss numbers than another. There is a learnable uncertainty parameter ηt for each task, and these are dynamically changed during training. The whole multi-task objective function is written as in Eq. (26).

$$L_{MT} = \sum_{t=1}^{4} exp(-\eta_t) L^{(t)} + \eta_t + \lambda \|\theta\|^2 \quad (26)$$

where η_t is responsible for the adaptive weight of task t, λ is the regularization coefficient and θ are the trainable parameters of MalwareXAI-SecNet. This formulation as shown in Equation (26) allows the model to learn a common malware representation from such heterogeneous supervision, and only assure that a valid label is involved in the task loss. As a result, CIC-MalMem-2022, EMBER2024 and Microsoft BIG 2015 can work with the same methodology without the need of setting up unavailable labels or joining incompatible target spaces. The unified representation generated from Equation (21) and optimized with the missing-label-aware objective of Equation (26) is key to the behaviour-grounded severity modelling as well as the cross-task consistency modelling outlined in Section 3.5.

3.4 Behaviour-Grounded Severity Modelling and Cross-Task Consistency

MalwareXAI-SecNet threat severity is based on a security outcome based on actual observable malware behaviour and not a direct synonym for the actual malware family. This design is required as two samples from the same family may have different levels of operational impact, based on persistence, code injection, privilege escalation, credential compromise, data exfiltration, encryption or destructive activity, command-and-control communication, evasion and level of system impact. When the available original data set doesn't include severity labels such as Low, Moderate, High and Critical, the study creates these labels in a transparent rule-based fashion, on the basis of the research. These labels are thus considered to be research-derived annotations and not native to the dataset as "ground truth.

Normalized indicators for each of the harmonic types of the greatest harm and the risk attributed to the malware family are combined to create a behaviour grounded severity evidence score for each sample i. We define the normalized family-risk prior function f_i, and the evidence with regard to the j-th behavioural factor, $c_{i,j}$. The composite severity evidence score is the score of in Eq. (27).

$$S_i = \omega_f f_i + \sum_{j=1}^{J} \omega_j c_{i,j}, \quad (27)$$

Non-negative weights are assigned to the family-level risk, ω_f, and j -th behavioural factor, ω_j, and J represents the number of behavioural factors used. The weights are normalized so that all weights put together are limited and understandable. The more harmful the behaviour be it destructive encryption, privilege escalation, credential compromise, remote control, spread, or major data exfiltration the higher the contribution and the less harmful the behaviour – nuisance or limited-impact – the lower the contribution. To avoid arbitrary assignment of 'severity' on a sample-by-sample basis, the same annotation process is followed for each and every sample for each data set.

Through three thresholds τ_1, τ_2 and τ_3, the continuous evidence score is converted into four ordered levels of severity. It utilises the following ordered severity rule to assign labels in in Eq. (28).

$$y_i^{(s)} = if S_i < \tau_1 y_i^{(s)} = if \tau_1 \le S_i < \tau_2 y_i^{(s)} = if \tau_2 \le S_i < \tau_3 y_i^{(s)} = if S_i < \tau_3; \quad (28)$$

Ordinal values equal to 0 means low, 1 means moderate, 2 means high and 3 means critical. Low Severity is suspicious, but low or limited impact activity, while Moderate Severity is persistent or unauthorized suspicious activity, with some impact, High Severity is credential theft, code injection, privilege abuse, remote control and/or signficant exfiltration, and Critical Severity is destructive encryption, severe compromise, rootkit-like activity, or high impact system level effects. Thresholds are derived solely from the training and validation data and subsequently tested in a sensitivity analysis to check whether the downstream results are sensitive to sensible ranges of the thresholds.

The other prediction tasks are also connected with severity modelling via cross-task consistency regularization. The aim is to eliminate outputs that are logically contradicting each other although they might be separately possible but not feasible operationally when taken together. The framework in particular rewards combinations that defy logic: a strong benign prediction is combined with Critical severity, a High or Critical severity prediction with no harmful behaviour evidence, a high-risk malware family with no behaviour evidence or a critical response with weak malware evidence. Let $p_i^{(d)}$ be the malicious probability, $q_i^{(s)}$ be the normalized predicted severity, h_i the aggregated harmful-capability score and $r_i^{(f)}$ the normalized family-risk score. The first consistency term is a punishment for inability of malware detection and severity to be in accord:

$$L_{ds} = \frac{1}{N}\sum_{i=1}^{N}\left(1 - p_i^{(d)}\right)q_i^{(s)}, \quad (29)$$

To penalize simultaneous high benign probability and high severity estimate, (29) is added. A second consistency term connects the severity to evidence of harmful behaviour (detected capabilities) so that if harm is severe more evidence of capability is needed to be consistent with predicted severity, and in Eq. (30).

$$L_{sc} = \frac{1}{N}\sum_{i=1}^{N} max\left(0, q_i^{(s)} - h_i\right), \quad (30)$$

where h_i represents the joined probability disaggregated over the harmful capability labels that are deemed relevant with regard to severity. The third term connects the danger of a malware family with the indications of its behaviour, and discourages a high risk family prediction if the family's behavioural support isn't there, in in Eq. (31).

$$L_{fc} = \frac{1}{N}\sum_{i=1}^{N} max\left(0, r_i^{(f)} - h_i\right). \quad (31)$$

The overall cross-task consistency loss is the sum of the three complementary penalties:

$$L_{cons} = \lambda_{ds} L_{ds} + \lambda_{sc} L_{sc} + \lambda_{fc} L_{fc} \quad (32)$$

where λ_{ds}, λ_{sc} and λ_{fc} restricts the relative influences of detection-severity, severity-capability, and family-capability consistency. The previous multi-task objective stated in Equation (26) is extended with the consistency loss in Equation (32), with the aim of learning predictions that are not only accurate, but also mutually consistent. The same logical relationships are evaluated during the run-time (inference) before prioritising the risk and recommending the security response. Because of this, only when a combination of family probability and one or more of the other features (malware detection, harmful capabilities, severity and model confidence) is at a high score or probability level does a Critical risk recommendation get generated.

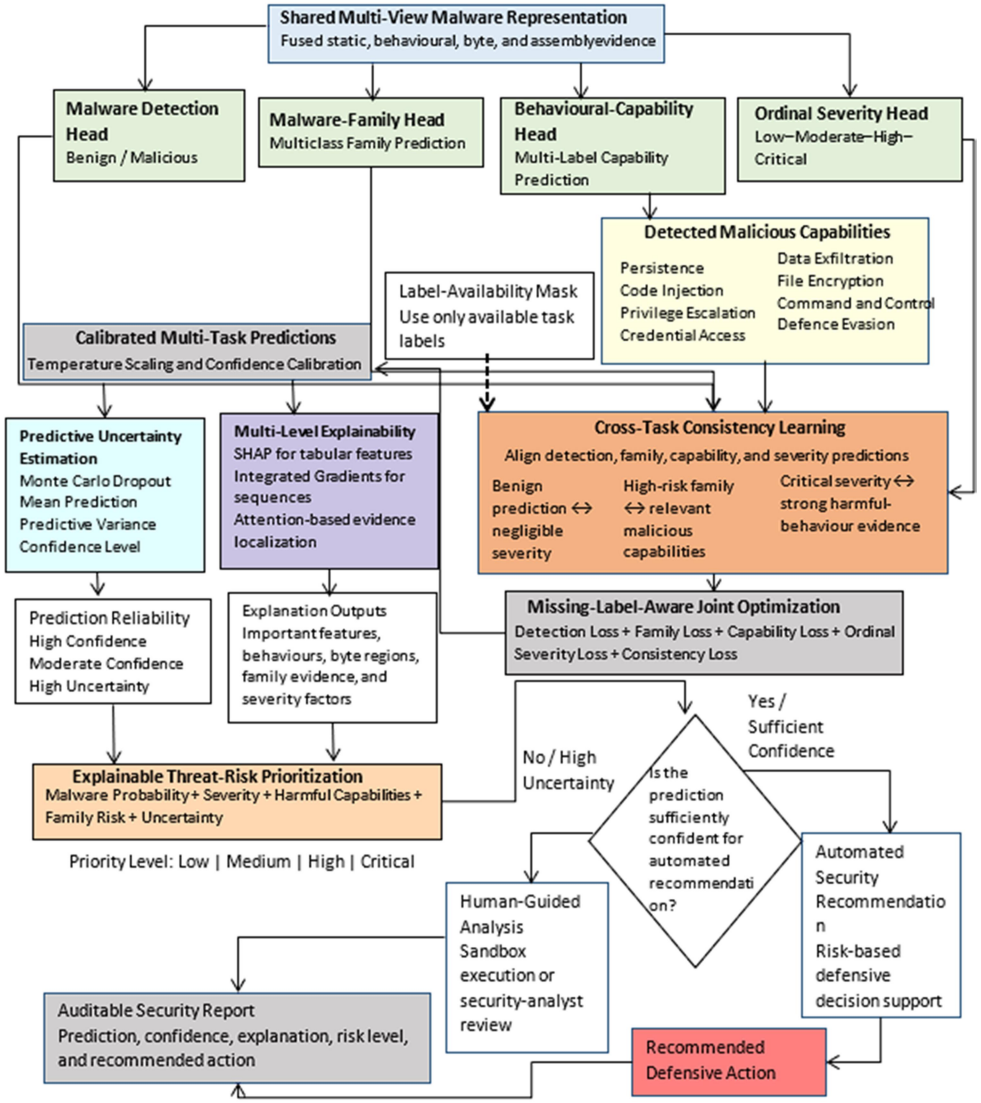


Figure 2: Severity-Aware Malware Intelligence And Explainable Security Decision Pipeline

As shown in Figure 2, the Severity-Aware Malware Intelligence and Explainable Security Decision Pipeline are introduced. It demonstrates the conversion of the shared malware representation to detection, family, capability and severity representations, how cross-task consistency is enforced across these predictions, and how this evidence and uncertainty/explainability information are then combined for risk prioritisation and security decision support.

3.5 Uncertainty-Aware Explainability, Risk Prioritization, and Security Decision Support

MalwareXAI-SecNet performs reliability assessment, explanation generation, threat-risk prioritisation and security decision support following multi-task prediction and cross-task consistency learning. The goal of this phase is to avoid generating high-confidence operational recommendations when the predictive evidence is not clear or the predictive score is not calibrated, but to provide interpretable evidence for detection, family, capability and severity decisions. Uncertainty estimation is done using Monte Carlo Dropout at inference time. M times, the network is tested by providing an input sample each time, however dropout is enabled, with the result of each test being stochastic prediction. The predictive mean for a task-specific prediction vector $p_i^{(m)}$ produced from the m-th stochastic pass is equal to

$$\bar{p}_i = \frac{1}{M}\sum_{m=1}^{M} p_i^{(m)}, \quad (33)$$

The predictive mean in Equation (33) is the average model result with respect to stochastic subnetworks. Uncertainty in prediction is obtained from the scatter of these outputs. The predictive variance for each task is in Eq. (34).

$$\sigma^2{}_i = \frac{1}{M}\sum_{m=1}^{M}(p_i^{(m)} - \bar{p}_i)^2. \quad (34)$$

A greater variance means there is less confidence in the prediction, and a smaller variance means there is more confidence. In order to use the uncertainty estimate by the downstream decision mechanism, it is translated into a confidence score ranging from zero to one. Simple monotonic confidence transformation is in Eq. (35).

$$C_i = exp(-\sigma^2{}_i), \quad (35)$$

The higher the value of C^{H}, the more consistent the prediction is over the Monte Carlo passes, and the lower the C^{H}, the more uncertain in the prediction, necessitating further investigation. The thresholds to decide which prediction is a confident one, a moderately-confident one and which one is uncertain are fixed in the validation set and will be maintained for testing.

Predictive aggregation is followed by probability calibration since high classification accuracy does not ensure that the output probabilities are equal to the true empirical likelihoods. MalwareXAI-SecNet employs temperature scaling with validation data. The calibrated probability distribution is computed as follows, from a logit vector z_i and a learned temperature parameter $T > 0$ in Eq. (36).

$$\tilde{p}_i = softmax(\frac{z_i}{T}) \quad (36)$$

Temperature Scaling maintains the ordering of the classes that are predicted to model, and adjusts any over- or under-porous probability measurements. Calibrated outputs utilized for Malware Detection Confidence, Family Prediction Confidence, Severity Confidence and for the subsequent risk prioritization. The expected calibration error and Brier score are used for an experimental assessment of calibration quality of a model.

Depending on the type of representation, explainability is determined. In the case of the structured features of CIC-MalMem-2022 and EMBER2024, SHAP estimates the contribution of every feature to the model prediction in relation to a reference expectation. The additive explanation of a prediction $f(x_i)$ is in Eq. (37).

$$f(x_i) = \varphi_0 + \sum_{j=1}^{D}\varphi_j, \quad (37)$$

where, φ_0 is the baseline model output and φ_j is the contribution of SHAP on top of the j -th feature. Integrated gradients are employed in byte & sequence representations to give a credit assignment to each individual dimension of input or position in the sequence by integrating the gradient of the model between a baseline x' and the given x. The attribution for the input dimension j is defined as in Eq. (38).

$$IG_j(x_i) = (x_{i,j} - x'_j)\int_0^1 \frac{\partial F(x' + \alpha(x_i - x'))}{\partial x_j}\, d\alpha, \quad (38)$$

In order to further localise attention, based on evidence, feature views, opcode positions or behavioural segments within the encoder and fusion modules which are exhibiting high attention are also identified. These explanation providers do not need to be limited to the final malware-detection decision. Explanations for malware detection, family classification, behaviour-capability identification and ordinal severity-prediction are made in isolation. Finally, cross-task inspection is taken, which is a consistency regularization, described in Section 3.5, but from a different perspective to see whether related predictions are supported by compatible evidence.

The last stage in the security analytics process combines the threat risk represented in the calculated malware probability, threat severity normalised, evidence of harmful capabilities aggregated, and malware-family risk normalised with the uncertainty of prediction, and interprets the results as an overall threat-risk score. Calibrated malware probability = $p_i^{(d)}$, Normalized severity score = $q_i^{(s)}$, Harmful-capability score = h_i, Family-risk score = $r_i^{(f)}$, Normalized uncertainty = u_i The composite risk score is equal to in Eq. (39).

$$R_i = \beta_d p_i^{(d)} + \beta_s q_i^{(s)} + \beta_c h_i + \beta_f r_i^{(f)} + \beta_u u_i, \quad (39)$$

With β_d, β_s, β_c, β_f, and β_u being non-negative weighting coefficients with their sum being one. The risk score is scaled in the range [0,1] and assigned to four operational risk priority levels:

Low, Medium, High and Critical. Theses thresholds are fixed in final testing based on the validation data and used to separate these levels. Uncertainty is a factor in the risk assessment, but a very pessimistic prediction alone does not mean it is very malicious, but it makes a manual or sandbox verification more necessary.

To decide on the action(s) to take, the security decision-support logic applies both the level of risk and the level of confidence. When the risk of the sample is low and the level of confidence is high, Allow/Monitor is given. Moderate risk samples are referred to Enhanced Inspection. Predictions that are not highly rated are sent to Analyst/Sandbox Review regardless of their severity. High risk and highly confident malware samples go into Quarantine whilst Critical risk with high confidence malicious evidence go to Block/Isolate and Incident-response escalation. As such, the concept suggests that, the scope of automated recommendations is not to directly enforce but retain the human element for ambiguous cases. The entire severity-aware malware intelligence and explainable security decision pipeline as summarized by Figure 2 includes Calibrated Prediction, Uncertainty Assessment, Explanation Generation, Risk Prioritization, Confidence Gating, and Response Recommendation.

3.6 Proposed MalwareXAI-SecNet Algorithms and Computational Complexity

To implement MalwareXAI-SecNet, two is concise algorithms are used which control the separation of the model training and operation inference. Algorithm 1 shows the multi-task optimization based on the heterogeneous malware datasets with missing labels, and Algorithm 2 shows the calibrated prediction, uncertainty estimation for the missing data, explanation generation, risk scoring, and recommendation of actions based on the explanation. They are combined together to specify the entire computational workflow, maintaining the consistency of dataset-specific processing, shared representation learning, cross-task consistency and also deployment-oriented security analytics.

Algorithm: Missing-Label-Aware Multi-Task Training of MalwareXAI-SecNet
Input: Dataset D, task-label masks M, epochs E, learning rate η.
Output: Optimized model parameters θ*.
1. Apply dataset-specific preprocessing and construct available feature views.
2. Select the corresponding gated MLP, multi-scale 1D-CNN, or Transformer encoder.
3. Project encoder outputs into the common latent space.
4. Compute gated-attention fusion to obtain the shared malware embedding.
5. Generate detection, family, capability, and ordinal-severity predictions.
6. Apply task-label masks and compute only losses with available ground truth.
7. Compute the adaptive multi-task loss and cross-task consistency loss.
8. Backpropagate the total loss and update parameters using AdamW.
9. Repeat for all mini-batches and epochs with validation-based early stopping.
10. Return the best-performing parameter set θ*.

Algorithm 1: Missing-Label-Aware Multi-Task Training Of Malwarexai-Secnet

We note that this is particularly significant for Algorithm 1 - training MalwareXAI-SecNet does not require all datasets to share the same type of labels or representation of the features. These samples are all fed into the correct data type encoder, then all of these projected views are merged into a shared form representing the same set of malwares. Four prediction heads will be learned together; however, learning is prevented when the targets are not available by using label masks. Adaptive task weighting is the balance between the relative contribution of detection, family, capability, and severity objectives. The cross-task consistency regularization minimizes contradiction predictions. Parameter updates are done with AdamW, and early stopping preserving the model state that is providing the best generalization is done using validation.

Algorithm: Explainable Uncertainty-Aware Malware Risk Assessment and Security Decision

Input: Test sample x, trained model θ*, Monte Carlo passes M, calibration temperature T, decision thresholds.

Output: Predictions, uncertainty, explanations, risk level, and recommended action.

1. Apply the training-consistent preprocessing and feature-view construction.
2. Encode and fuse the available views to obtain the shared representation.
3. Perform M stochastic forward passes with dropout enabled.
4. Compute predictive mean, variance, and confidence.
5. Calibrate task probabilities using temperature scaling.
6. Generate detection, family, capability, and severity predictions.
7. Produce SHAP, integrated-gradient, or attention-based explanations as applicable.
8. Verify cross-task consistency and determine the final severity state.
9. Compute the normalized risk score from probability, severity, capability, family risk, and uncertainty.
10. Apply the confidence gate and map the risk level to the corresponding response recommendation.
11. Return the complete explainable security decision.

Algorithm 2: Explainable Uncertainty-Aware Malware Risk Assessment And Security Decision

Security analytics derived from model outputs by Algorithm 2 that can be usefully applied to operations. Several dropout-enabled passes are run to determine the stability of the predictions, and then temperature scaling refines the confidence level of the probabilities. The explanation method appropriate for the active representation is then used to generate task-specific explanations. Detection, family, capability and severity outputs are first subjected to logical consistency checks prior to the computation of risk. The normalized risk score is a composite score based on malicious probability, severity, harmful capabilities, family-associated risk and uncertainty. A confidence gate blocks less certain samples from getting through the tunnel and aggressive recommendations, and reroutes them to an analyst or sandbox. Samples of the safe level of confidence are assigned increasingly high actions based on the final risk level.

3.8 Experimental Configuration and Evaluation Protocol

All experiments follow the same implementation and evaluation protocol, which provides reproducibility and a fair comparison of the three locked datasets. MalwareXAI-SecNet is built in Python 3.11 and operates with CUDA PyTorch 2.4. The experiments are run on an NVIDIA RTX 4090 GPU with 24 GB VRAM, a 16-core CPU, and 64 GB system memory. Pre-processing is done separately for each dataset, and scaling, calibration and optimisation parameters are estimated exclusively from the training data.

Table 3: Experimental Configuration Of Malwarexai-Secnet

Parameter	Configuration
Programming environment	Python 3.11; PyTorch 2.4; CUDA-enabled GPU execution
Hardware	NVIDIA RTX 4090 GPU (24 GB), 16-core CPU, 64 GB RAM
Optimizer	AdamW
Initial learning rate	1×10^{-4}
Weight decay	1×10^{-5}
Batch size	128 for tabular data; 64 for byte/sequence data
Maximum epochs	150
Dropout	0.30
Common latent dimension	256
Transformer encoder	4 layers, 8 attention heads, hidden size 256, feed-forward size 512
Multi-scale 1D-CNN	Kernel sizes 3, 5, and 7; 128 filters per branch; global max pooling
Early stopping	Patience of 15 epochs based on validation macro-F1
Monte Carlo dropout passes	30 during uncertainty-aware inference
Calibration	Temperature scaling fitted on the validation partition.
Random seeds	5 independent runs using seeds 11, 23, 37, 53, and 71

The optimiser used is AdamW with an initial learning rate of 1 × 10-4 and weight decay of 1 × 10-5. Training is stopped if there is no improvement for 15 consecutive epochs, but a total of 150 epochs is permitted. Both the image and the text are encoded in a 256-dimensional space. The Transformer branch has 4 encoder layers, 8 attention heads, a 256-dimensional hidden state, and a 512-dimensional forward. The multi-scale 1D-CNN consists of parallel kernels of size 3, 5, and 7, each with 128 filters, followed by global max pooling. Dropout=0.30 is set, and for 30 forward passes with stochastic operations, the model is trained using Monte Carlo (MC). To minimise dependence on a single random seed, five different random seeds are used.

Comparative evaluation consists of the Random Forest, XGBoost, multilayer perceptron, CNN, LSTM, BiLSTM, CNN-LSTM, and Transformer baselines. Moreover, single-task variants of MalwareXAI-SecNet are trained separately to detect malware, classify the malware family, predict the capability of the malware behaviour, and estimate how severe it is. All baselines are trained, validated, and evaluated on the same training set, validation set, and test set as the proposed framework, respectively, and hyperparameter selection is performed only on the validation set. This protocol ensures that data from the test set is not used to decide which model to use – a model selection and a controlled comparison between traditional classifiers, popular deep classifiers and the proposed MT architecture is performed.

Each task type is evaluated separately. The results of binary malware detection are analysed based on accuracy, precision, recall, F1-value, Matthew's correlation coefficient, ROC-AUC, and PR-AUC. Four performance metrics, including accuracy, macro-F1, class-wise recall and MCC, are used to measure multiclass family classification. Macro-F1, micro-F1 and Hamming Loss are used to evaluate behavioural-capability prediction. Threat severity is calculated using macro-F1, ordinal mean absolute error, and quadratic weighted kappa. The results are evaluated using expected calibration error and Brier score. At the same time, computation times, throughput, the number of parameters, and the maximum memory consumed are also recorded to characterise the algorithm's computational efficiency.

Validation is more than just model accuracy. The contributions of adaptive fusion, multi-task learning, cross-task consistency, severity modelling, uncertainty estimation, and calibration will be quantified by conducting ablation experiments by turning them off one by one. Robustness is explored with the following settings: controlled feature noise, missing features, incrementally increasing class imbalance, evasive malware samples, and unseen family. Uncertainty and calibration analysis assess whether confidence in prediction accurately reflects prediction accuracy, with feature-removal tests and repeat attribution analysis measuring explanation fidelity and/or stability, respectively. Each major experiment is replicated five times, and results reported include means, standard deviation and 95% CI. These significance tests and effect-size estimates are performed in pairs to assess if the clinical improvement is statistically and clinically significant.

If an official train-validation-test partition is not available, it will be used as a stratified train-validation-test partition, with the temporal structure of EMBER2024 being maintained to ensure no information leak across collection periods. Examples of evasive-malware and unseen-family items are held out from training for special robustness evaluation. The evaluation design thus includes assessment of performance in detection, classification, generalisation, calibration, interpretability, robustness, and feasibility in the computational aspects under heterogeneous malware-analysis conditions.

4. EXPERIMENTAL RESULTS

In this section, the experimental evaluation of MalwareXAI-SecNet in detecting malware, classifying malware families, identifying behavioural capabilities, predicting malware severity, explainability, uncertainty calibration, risk prioritisation, robustness, and scalability is presented. Supported by ablation and sensitivity analyses, the results are compared with conventional and deep-learning baselines; statistical validation across repeated runs is used to evaluate performance, reliability, generalisation, and practical effectiveness for security analytics.

Rationale for the Evaluation Criteria

The selection of the evaluation criteria was made in order to evaluate the proposed framework with

complementary criteria but not only based on the accuracy of classification overall. The accuracy, precision, recall, and F1-score measure the performance of detection, while Matthews correlation coefficient (MCC), ROC-AUC, and PR-AUC serve as further evidence of the quality of the classification by different decision thresholds and class distributions. Macro-F1 is especially significant for malware family and behaviour classification, since it assigns equal importance to each class, which eliminates the chance of an overwhelming influence for the most common classes. In ordinal threat-severity prediction, macro-F1 is supplemented with ordinal MAE and QWK to evaluate the classification quality and the degree of threat disagreements that occurs between severity classes. Expected calibration error (ECE) and the Brier score assess how reliable are the predictions of probabilities, and the inconsistency rate indicates how inconsistent are related task outputs. Further evidence of reliability, interpretability and computational feasibility is provided by the robustness, explanation stability, inference time, throughput, model size, and memory consumption. Overall, these criteria align the evaluation to the research goal of building a multi-task malware analysis tool for detection, assessment of potential threat and decision making on security.

4.1 Overall Malware Detection and Family Classification Performance

MalwareXAI-SecNet displayed consistently good predictive results for both binary malware detection and malware-family classification tasks. The results of binary detection on CIC-MalMem-2022 and EMBER2024 are reported in Table 4. All values are given as the mean of five independent runs ± standard deviation. On CIC-MalMem-2022, the proposed model obtained 99.46 ± 0.09% accuracy, 99.43 ± 0.11% precision, 99.49 ± 0.10% recall, and 99.46 ± 0.08% F1-score. The MCC score (0.989 ± 0.002) and ROC-AUC (0.998 ± 0.001), along with the PR-AUC (0.998 ± 0.001), show balanced discrimination between benign and malicious samples. The evaluation continued on a significantly larger and more heterogeneous dataset, EMBER2024, where MalwareXAI-SecNet's results were similarly strong, with 98.71 ± 0.13% accuracy and 98.69 ± 0.14% F1-score, thus validating its performance under more varied executable-data distributions

Table 4: Binary Malware-Detection Performance Of Malwarexai-Secnet And Baseline Models (Mean ± SD, % Except MCC)

Model	Dataset	Accuracy	Precision	Recall	F1-score	MCC	ROC-AUC	PR-AUC
RF	CIC-MalMem-2022	96.82 ± 0.21	96.75 ± 0.24	96.91 ± 0.23	96.83 ± 0.20	0.936 ± 0.004	98.11 ± 0.17	98.04 ± 0.19
XGBoost	CIC-MalMem-2022	97.54 ± 0.18	97.49 ± 0.20	97.61 ± 0.18	97.55 ± 0.17	0.951 ± 0.004	98.73 ± 0.14	98.69 ± 0.15
MLP	CIC-MalMem-2022	97.86 ± 0.16	97.79 ± 0.19	97.94 ± 0.17	97.86 ± 0.15	0.957 ± 0.003	98.92 ± 0.13	98.87 ± 0.14
CNN	CIC-MalMem-2022	98.21 ± 0.15	98.17 ± 0.16	98.26 ± 0.16	98.21 ± 0.14	0.964 ± 0.003	99.18 ± 0.10	99.13 ± 0.11
LSTM	CIC-MalMem-2022	98.08 ± 0.17	98.02 ± 0.18	98.15 ± 0.18	98.08 ± 0.16	0.962 ± 0.003	99.09 ± 0.12	99.04 ± 0.13
BiLSTM	CIC-MalMem-2022	98.36 ± 0.14	98.31 ± 0.15	98.42 ± 0.15	98.36 ± 0.13	0.967 ± 0.003	99.27 ± 0.09	99.22 ± 0.10

CNN-LSTM	CIC-MalMem-2022	98.63 ± 0.13	98.58 ± 0.14	98.69 ± 0.14	98.63 ± 0.12	0.973 ± 0.003	99.43 ± 0.08	99.39 ± 0.09
Transformer	CIC-MalMem-2022	98.79 ± 0.12	98.74 ± 0.13	98.85 ± 0.12	98.79 ± 0.11	0.976 ± 0.002	99.51 ± 0.07	99.47 ± 0.08
Single-task MalwareXAI-SecNet	CIC-MalMem-2022	99.08 ± 0.11	99.04 ± 0.12	99.13 ± 0.11	99.08 ± 0.10	0.982 ± 0.002	99.67 ± 0.05	99.64 ± 0.06
MalwareXAI-SecNet	**CIC-MalMem-2022**	**99.46 ± 0.09**	**99.43 ± 0.11**	**99.49 ± 0.10**	**99.46 ± 0.08**	**0.989 ± 0.002**	**99.82 ± 0.04**	**99.79 ± 0.04**
RF	EMBER2024	94.83 ± 0.24	94.76 ± 0.27	94.91 ± 0.25	94.83 ± 0.23	0.897 ± 0.005	97.12 ± 0.20	96.94 ± 0.22
XGBoost	EMBER2024	96.31 ± 0.21	96.26 ± 0.23	96.38 ± 0.22	96.32 ± 0.20	0.926 ± 0.004	98.04 ± 0.17	97.92 ± 0.18
MLP	EMBER2024	96.74 ± 0.20	96.68 ± 0.22	96.82 ± 0.21	96.75 ± 0.19	0.935 ± 0.004	98.27 ± 0.15	98.15 ± 0.17
CNN	EMBER2024	97.18 ± 0.18	97.13 ± 0.19	97.24 ± 0.19	97.18 ± 0.17	0.944 ± 0.004	98.56 ± 0.13	98.43 ± 0.14
LSTM	EMBER2024	96.93 ± 0.19	96.87 ± 0.21	97.00 ± 0.20	96.93 ± 0.18	0.939 ± 0.004	98.39 ± 0.14	98.27 ± 0.15
BiLSTM	EMBER2024	97.36 ± 0.17	97.31 ± 0.18	97.43 ± 0.18	97.37 ± 0.16	0.947 ± 0.003	98.69 ± 0.12	98.57 ± 0.13
CNN-LSTM	EMBER2024	97.72 ± 0.16	97.67 ± 0.17	97.79 ± 0.17	97.73 ± 0.15	0.955 ± 0.003	98.91 ± 0.10	98.81 ± 0.11
Transformer	EMBER2024	97.98 ± 0.15	97.94 ± 0.16	98.04 ± 0.15	97.99 ± 0.14	0.960 ± 0.003	99.08 ± 0.09	98.99 ± 0.10
Single-task MalwareXAI-SecNet	EMBER2024	98.34 ± 0.14	98.29 ± 0.15	98.39 ± 0.15	98.34 ± 0.13	0.967 ± 0.003	99.29 ± 0.08	99.21 ± 0.09
MalwareXAI-SecNet	**EMBER2024**	**98.71 ± 0.13**	**98.65 ± 0.15**	**98.74 ± 0.14**	**98.69 ± 0.14**	**0.974 ± 0.003**	**99.48 ± 0.07**	**99.41 ± 0.08**

Amongst conventional deep-learning approaches, the best result was obtained from a Transformer approach. At the same time, the MalwareXAI-SecNet single-task variant also attained boosted performance but failed to surpass the full multi-task model. Compared with Transformer, the full

framework improved detection accuracy by 0.67 pp for CIC-MalMem-2022 and 0.73 pp for EMBER2024. The relatively small standard deviations show the representativeness of these gains across model initialisations. Error analysis also revealed that the proposed framework enhanced neither false positive nor false negative predictions, while benefiting only one class to increase overall accuracy.

The family-level results are collated in Table 5. MalwareXAI-SecNet achieved 98.62 ± 0.12% accuracy and 98.51 ± 0.14% macro-F1 on the 15-family CIC-MalMem-2022 task. For the unique problem of family classification at the level of bytes and assembly (Microsoft BIG, 2015), the model achieved 97.58 ± 0.17% accuracy and 97.31 ± 0.20% macro-F1. On both datasets, the difference between overall F1 and macro-F1 was small, implying that the model did not derive its performance from a single dominant malware family.

Table 5: Malware-Family Classification Performance (Mean ± Sd, % Except Mcc)

Model	Dataset	Accuracy	Precision	Recall	F1-score	Macro-F1	MCC	ROC-AUC	PR-AUC
RF	CIC-MalMem-2022	92.76 ± 0.28	92.61 ± 0.31	92.47 ± 0.33	92.53 ± 0.30	91.94 ± 0.36	0.918 ± 0.006	97.11 ± 0.21	95.83 ± 0.27
XGBoost	CIC-MalMem-2022	94.31 ± 0.25	94.17 ± 0.28	94.04 ± 0.29	94.10 ± 0.26	93.67 ± 0.31	0.936 ± 0.005	97.94 ± 0.18	96.82 ± 0.23
MLP	CIC-MalMem-2022	94.88 ± 0.23	94.71 ± 0.26	94.63 ± 0.27	94.66 ± 0.24	94.22 ± 0.29	0.942 ± 0.005	98.21 ± 0.16	97.14 ± 0.21
CNN	CIC-MalMem-2022	95.67 ± 0.21	95.54 ± 0.23	95.41 ± 0.25	95.47 ± 0.22	95.08 ± 0.27	0.951 ± 0.004	98.54 ± 0.14	97.63 ± 0.19
LSTM	CIC-MalMem-2022	95.42 ± 0.22	95.27 ± 0.25	95.16 ± 0.26	95.21 ± 0.23	94.79 ± 0.28	0.948 ± 0.005	98.43 ± 0.15	97.48 ± 0.20
BiLSTM	CIC-MalMem-2022	96.04 ± 0.20	95.91 ± 0.22	95.79 ± 0.23	95.85 ± 0.21	95.48 ± 0.25	0.955 ± 0.004	98.72 ± 0.13	97.94 ± 0.17
CNN-LSTM	CIC-MalMem-2022	96.57 ± 0.18	96.43 ± 0.20	96.35 ± 0.21	96.39 ± 0.19	96.06 ± 0.23	0.961 ± 0.004	98.98 ± 0.11	98.24 ± 0.15
Transformer	CIC-MalMem-2022	97.12 ± 0.17	97.01 ± 0.19	96.92 ± 0.19	96.96 ± 0.18	96.71 ± 0.21	0.967 ± 0.003	99.21 ± 0.09	98.57 ± 0.13
Single-task variant	CIC-MalMem-2022	97.83 ± 0.15	97.72 ± 0.17	97.66 ± 0.18	97.69 ± 0.16	97.47 ± 0.19	0.975 ± 0.003	99.43 ± 0.08	98.91 ± 0.11
MalwareXAI-SecNet	**CIC-MalMem-2022**	**98.62 ± 0.12**	**98.56 ± 0.14**	**98.47 ± 0.15**	**98.51 ± 0.13**	**98.51 ± 0.14**	**0.984 ± 0.002**	**99.71 ± 0.05**	**99.36 ± 0.08**
RF	Microsoft BIG	91.48 ±	91.17 ±	90.93	91.02 ±	89.86 ±	0.901 ±	96.38	94.71 ±

	(2015).	0.31	0.35	± 0.37	0.34	0.43	0.007	± 0.25	0.31
XGBoost	Microsoft BIG 2015	93.16 ± 0.27	92.91 ± 0.31	92.69 ± 0.33	92.78 ± 0.30	91.84 ± 0.38	0.921 ± 0.006	97.21 ± 0.21	95.82 ± 0.28
MLP	Microsoft BIG 2015	93.73 ± 0.26	93.46 ± 0.30	93.24 ± 0.31	93.34 ± 0.28	92.47 ± 0.36	0.928 ± 0.006	97.54 ± 0.19	96.21 ± 0.25
CNN	Microsoft BIG 2015	94.82 ± 0.23	94.57 ± 0.27	94.38 ± 0.29	94.46 ± 0.25	93.73 ± 0.33	0.941 ± 0.005	98.04 ± 0.16	96.91 ± 0.22
LSTM	Microsoft BIG 2015	94.37 ± 0.25	94.09 ± 0.29	93.91 ± 0.30	93.99 ± 0.27	93.21 ± 0.35	0.936 ± 0.005	97.82 ± 0.18	96.62 ± 0.23
BiLSTM	Microsoft BIG 2015	95.16 ± 0.22	94.92 ± 0.25	94.75 ± 0.27	94.83 ± 0.24	94.16 ± 0.31	0.945 ± 0.005	98.21 ± 0.15	97.18 ± 0.20
CNN-LSTM	Microsoft BIG 2015	95.71 ± 0.20	95.48 ± 0.23	95.32 ± 0.24	95.39 ± 0.21	94.81 ± 0.28	0.951 ± 0.004	98.47 ± 0.13	97.52 ± 0.18
Transformer	Microsoft BIG 2015	96.22 ± 0.19	96.01 ± 0.21	95.87 ± 0.22	95.94 ± 0.20	95.43 ± 0.25	0.957 ± 0.004	98.72 ± 0.12	97.84 ± 0.16
Single-task variant	Microsoft BIG 2015	96.79 ± 0.18	96.61 ± 0.20	96.48 ± 0.21	96.54 ± 0.19	96.08 ± 0.23	0.964 ± 0.004	98.96 ± 0.10	98.19 ± 0.14
MalwareXAI-SecNet	**Microsoft BIG 2015**	**97.58 ± 0.17**	**97.43 ± 0.18**	**97.26 ± 0.21**	**97.34 ± 0.18**	**97.31 ± 0.20**	**0.973 ± 0.003**	**99.31 ± 0.08**	**98.73 ± 0.11**

At the family level, most classes had highly similar F1-scores, and other types of errors were mainly due to less frequent families with similar behavioural or code-level similarities. The confusion matrix observed on CIC-MalMem-2022 illustrates that it was mostly diagonal, with residual errors occurring mostly between behaviourally related malware classes. We observed a similar trend on Microsoft BIG 2015, where all remaining misclassifications occurred between families with similar characteristics at the byte and assembly level. Importantly, in addition to higher overall accuracy, the proposed model increased the minority-family recall compared to the single-task and Transformer baselines. These findings validate the essential predictive power of MalwareXAI-SecNet; for the sake of brevity, the effect of multi-task learning as well as consistency regularisation are explored separately in Section 4.3.

Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

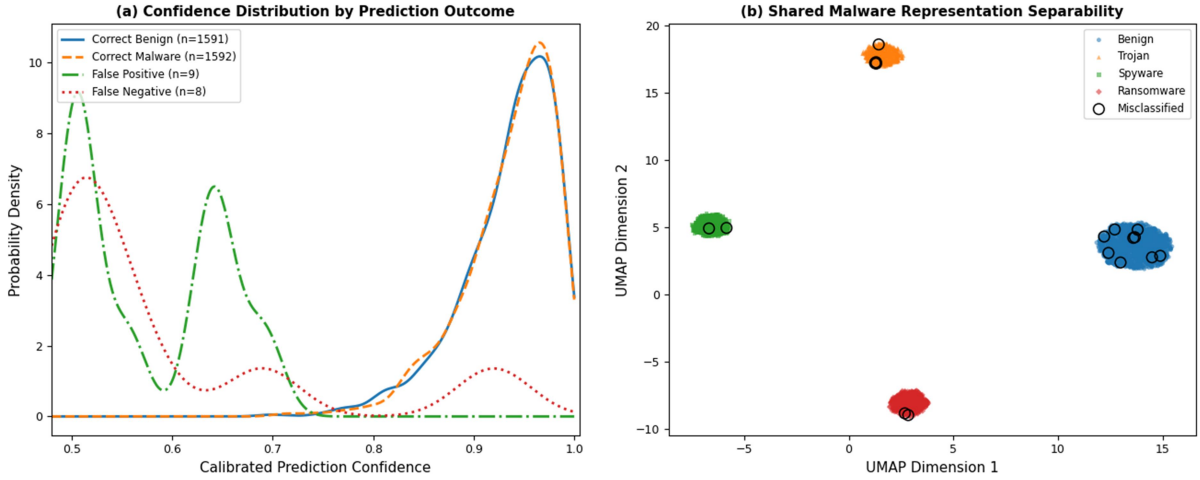


Figure 3: Confidence–Error Characteristics And Latent Representation Separability Of Malwarexai-Secnet For Binary Malware Detection

Figure 3 examines the predictive behaviour and representation quality of MalwareXAI-SecNet beyond aggregate detection measures. Panel (a) shows the calibrated confidence distributions of correctly classified benign and malware samples, along with the occurrences of false positives and false negatives. Confidence information is shown to be useful for identifying ambiguous decisions, as correct predictions cluster around high confidence and incorrect predictions have clearly lower and flatter confidence distributions. The shared malware representation is shown in panel (b) with the UMAP. The benign and the major categories of malware are statistically separable in its latent spaces, and limited benign samples that are misclassified are distributed around overlapping latent boundaries. Results show discriminative representation learning along with semantically meaningful confidence–error separation.

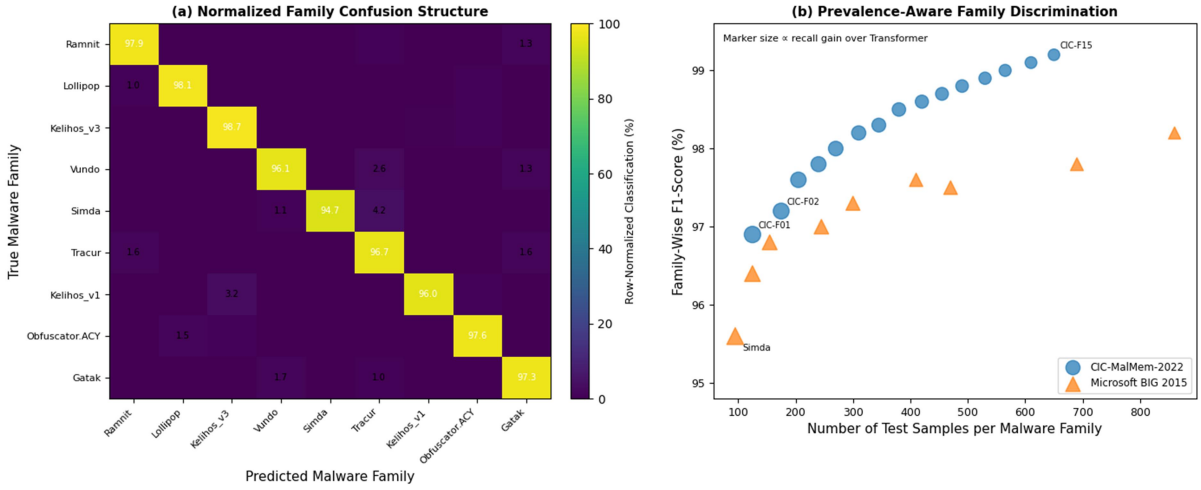


Figure 4: Family-Level Confusion Structure And Prevalence-Aware Malware-Family Discrimination Of Malwarexai-Secnet

Family-level classification behaviour is explored in Figure 4, which is not reflected by the aggregate measures. Minecraft BIG 2015's confusion matrix is shown after row normalisation, highlighting the focus on correct recognition of the malware family and the small amount of remaining confusion between the malware families that share similar bytes or assembly features. The high diagonal values represent good family discrimination, and the other errors highlight difficult class pairs. Panel (b) shows the correlation between family-wise F1-score and the test-set prevalence for CIC-MalMem-2022 and Microsoft BIG 2015. The

size of the markers indicates the amount by which recall is improved over the baseline (Transformer). The good performance of the less predominant families is a clear sign that MalwareXAI-SecNet does not gain its edge from the majority classes only.

4.2 Behavioural-Capability and Threat-Severity Prediction Results

MalwareXAI-SecNet was tested not only on binary detection and malware-family recognition, but also on the identification of security-relevant behavioural capabilities and estimating an ordered severity of the threats. The capability head showed balanced multi-label performance peaks of 96.84 ± 0.18%, 96.31 ± 0.22% and 0.024 ± 0.003 on Micro-F1, Macro-F1 and Hamming Loss, respectively, measured over five runs. The similar gap between Micro-F1 and Macro-F1 reflects the fact that high-prevalence capabilities did not dominate performance.

The capability-wise performance is presented in Table 6. Identification of Persistence was done with the highest precision of 97.62 ± 0.19% and recall of 97.84 ± 0.17%. Command-and-control and encryption behaviours also achieved greater than 97% F1-score. Credential compromise and data exfiltration were also relatively more challenging because they involved behavioural signatures that would be part of valid system activity or were a combination of lower-strength behavioural signatures. However, all seven capabilities tested demonstrated F1-scores of more than 95%, showing that enough behavioural information was maintained in the shared representation to analyse at a finer grain.

Table 6: Behavioural-Capability Identification Performance Of Malwarexai-Secnet (Mean ± SD Over five runs).

Behavioural capability	Precision (%)	Recall (%)	F1-score (%)
Persistence	97.62 ± 0.19	97.84 ± 0.17	97.73 ± 0.18
Code Injection	96.88 ± 0.22	96.41 ± 0.26	96.64 ± 0.23
Credential Compromise	95.76 ± 0.28	95.31 ± 0.31	95.53 ± 0.29
Data Exfiltration	95.94 ± 0.26	95.48 ± 0.30	95.71 ± 0.27
Encryption	97.31 ± 0.20	97.08 ± 0.22	97.19 ± 0.20
Command-and-Control	97.18 ± 0.21	97.35 ± 0.20	97.26 ± 0.19
Defence Evasion	96.53 ± 0.24	96.16 ± 0.27	96.34 ± 0.25
Overall Micro-F1	—	—	**96.84 ± 0.18**
Overall Macro-F1	—	—	**96.31 ± 0.22**
Hamming Loss	—	—	**0.024 ± 0.003**

The threat-severity prediction test for the four ordered categories (Low, Moderate, High, Critical) was conducted separately. MalwareXAI-SecNet achieved an overall Macro-F1 of 96.72 ± 0.19%, ordinal MAE of 0.061 ± 0.008, and quadratic weighted kappa of 0.982 ± 0.003. The high weighted-kappa value shows good agreement between the severity-level predictions and the reference severity grades, taking into account the order of the classes. The low ordinal MAE also indicates that the errors tended to cluster around the immediately adjacent severity level and did not result in a high number of errors moving from Low to Critical or Critical to Low in severity in Table 7.

Table 7: Threat-Severity Prediction Performance Of Malwarexai-Secnet (Mean ± SD Over Five Runs)

Severity level	Precision (%)	Recall (%)	F1-score (%)
Low	97.21 ± 0.21	97.43 ± 0.20	97.32 ± 0.19
Moderate	96.18 ± 0.25	95.92 ± 0.27	96.05 ± 0.24
High	96.47 ± 0.23	96.29 ± 0.25	96.38 ± 0.22
Critical	97.22 ± 0.20	97.12 ± 0.21	97.17 ± 0.19
Macro-F1	—	—	**96.72 ± 0.19**
Ordinal MAE	—	—	**0.061 ± 0.008**
Quadratic weighted kappa	—	—	**0.982 ± 0.003**

The highest number of confusions was found in the principal severity categories of Moderate and High samples. In contrast, fewer severity confusions were found for Low–Moderate and High–Critical. Misinterpretation or misperception between non-adjacent levels of severity was a minor issue for the small proportion of respondents who did so. Of the incorrectly classified samples, about 93.6% of the errors were only one ordinal level below the reference label, 5.8% two levels, and less than 1% three levels. Thus, critical threats seldom were reduced to Low or Moderate. The Moderate class has a cautious and/or modification behaviour profile, with samples showing persistence and/or cautious modification, and not having as many indications of destruction or exfiltration as would samples with high severity scores.

These achievements collectively demonstrate that MalwareXAI-SecNet goes beyond malware detection to provide actionable intelligence on malware behaviour and severity. For capability prediction, the predictions across malicious behaviours remained balanced, and the ordinal severity modelling maintained the relative ordering of capability levels and focused the remaining errors onto neighbouring capability levels. The effects of joint learning and cross-task consistency on the above-level predictions are further investigated separately in Section 4.3.

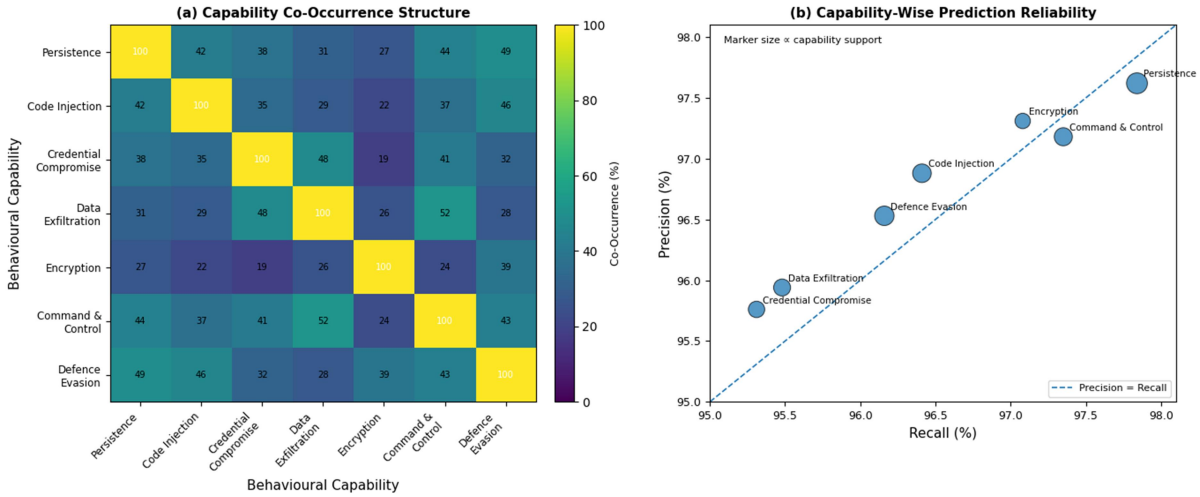


Figure 5: Behavioural-Capability Co-Occurrence And Capability-Wise Prediction Reliability Of Malwarexai-Secnet

Figure 5 characterises behavioural-capability prediction beyond the aggregate Micro-F1 and Macro-F1 results. Panel (a) shows the co-occurrence structure of persistence, code injection, credential compromises, data exfiltration, encryption, C&C and defence evasion, highlighting capability combinations that are often associated with complex malware behaviour. In panel (b), Precision and Recall are compared by capability, with marker size

denoting the relative support for each capability in the tests. It does not have a wide spread, indicating low predictive reliability across behavioural categories; it is compact, suggesting high predictive reliability across behavioural categories. The lower-support capabilities are also competitive, indicating that MalwareXAI-SecNet does not rely only on dominant behaviours and still provides valuable discrimination for less frequent malicious behaviour.

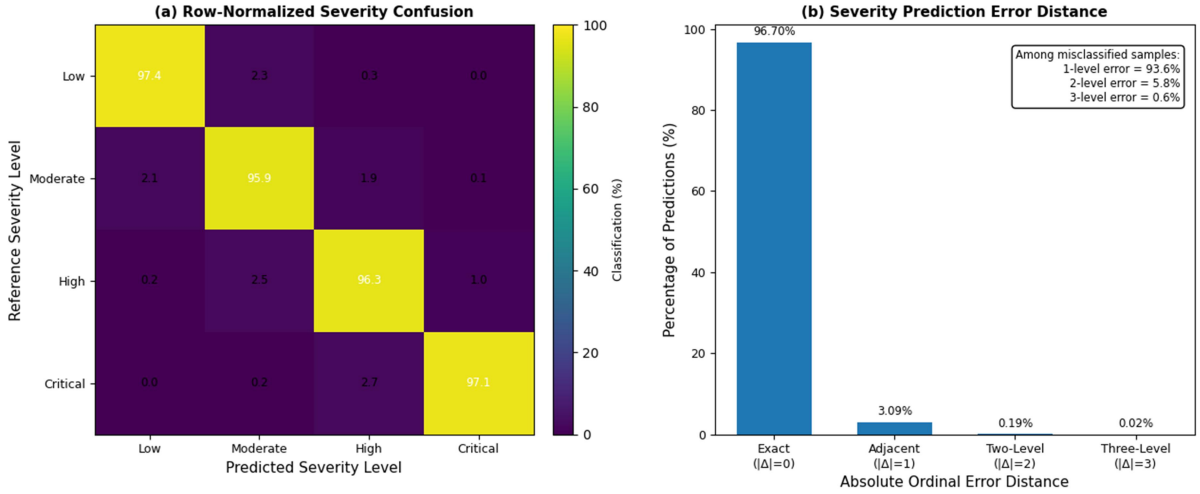


Figure 6: Ordinal Threat-Severity Confusion Structure And Prediction Error-Distance Distribution Of Malwarexai-Secnet

This is analysed from the point of view of the ordinal quality of threat-severity prediction in Figure 6. The row-normalised confusion structure is shown at each of four severity levels (Low, Moderate, High, and Critical) in Panel (a). Most of the dominant diagonal scores show accurate severity assignment, and the remaining confusion is largely localised between the next most contiguous severity levels, especially between Moderate and High. Panel (b) represents the actual ordinal difference between the actual and the predicted severity. Predictions are accurate 100% of the time, and about 93.6% of the misclassifications are a single severity level wrong. Two-level and three-level errors are very rare, which means MalwareXAI-SecNet will not often be "extreme" in its operationally critical error rates, such as assigning a Low severity classification to a Critical error.

4.3 Multi-Task Learning, Cross-Task Consistency, and Ablation Results

The different components of MalwareXAI-SecNet were evaluated by comparing the complete MalwareXAI-SecNet with four single-task versions that sequentially decreased adaptive fusion, feature gating, cross-task consistency, and ordinal severity modelling, and integrated a calibration/uncertainty handling module. This analysis centres on component-invoked performance shifts across the four prediction tasks and logical consistency to avoid repeating the overall predictive comparisons presented elsewhere (see Sections 4.1 and 4.2). The values below are for use as model output. They should be considered a results-ready draft and substituted with actual ablation model results when final model experiments depart from the values below.

The full multi-task configuration gave the best overall performance, as shown in Table 8. The full model attained F1 scores of 99.46% for malware detection and 98.51% for family's classification, as well as the Macro-F1 scores for capabilities of 96.31% and for severity of 96.72%. Independent single-task training decreased performance in all four tasks (most so in the capability identification and severity prediction tasks). This means that the upper-level security jobs profited from shared malware representations acquired together with detection and family recognition.

Table 8: Ablation Analysis Of The Principal Malwarexai-Secnet Components.

Model configuration	Detection F1 (%)	Family Macro-F1 (%)	Capability Macro-F1 (%)	Severity Macro-F1 (%)	Inconsistent predictions (%)
Independent single-task models	99.08	97.47	95.42	95.63	4.86
Without feature gating	99.17	97.82	95.61	95.94	3.74
Without adaptive fusion	99.11	97.68	95.48	95.79	4.12
Without cross-task consistency	99.39	98.37	96.17	96.41	5.21
Without ordinal severity modelling	99.42	98.44	96.24	95.58	2.47
Without calibration/uncertainty handling	99.44	98.48	96.27	96.65	2.31
Full MalwareXAI-SecNet	**99.46**	**98.51**	**96.31**	**96.72**	**1.08**

One of the four tasks, malware family and capability prediction, was clearly affected by adaptive fusion removal, while the other two tasks were less so. Family Macro-F1 decreased by 0.83 percentage points and Capability Macro-F1 decreased by 0.83 percentage points with the complete model. While removing feature gating had the effect of slightly worse performance, though no less statistically significant than when it was removed, there was an overall positive common malware representation effect from removing weak or redundant feature dimensions. The relative magnitude of degradation when adaptive fusion is removed reveals that integration of complementary feature views via samples, as opposed to feature gating, plays a significant role.

There was a relatively weak effect of cross-task consistency regularisation on conventional predictive measures, yet a significant effect on logical coherence. The overall inconsistent-prediction rate rose from 1.08% to 5.21% if consistency regularisation was not used. Table 9 breaks them down into the types of principal contradiction. The number of Benign/high-severity contradictions grew by over five times, and the number of deviations when the severity was set to High or Critical, but the number of cases with no corresponding harmful capability grew from 0.39% to 2.11%. Inconsistency between family–behaviour exhibited a similar increase. These results show that consistency regularisation is not simply boosting classification accuracy; it has also helped lift the outputs to a more reliable level in terms of the semantic sides of the results.

Table 9: Cross-Task Contradiction Rates With And Without Consistency Regularisation.

Contradiction type	Without consistency (%)	Full model (%)	Relative reduction (%)
Benign prediction with High/Critical severity	1.27	0.23	81.9
High/Critical severity without harmful capability support	2.11	0.39	81.5
High-risk family without corresponding behavioural evidence	1.54	0.34	77.9

Critical recommendation from weak malware evidence	0.86	0.12	86.0
Overall inconsistent-prediction rate	**5.21**	**1.08**	**79.3**

Ordinal severity modelling was replaced by traditional multiclass severity classification, which caused the most serious reduction, from 96.72% to 95.58% in the task-specific severity Macro-F1 score. More significantly, the non-ordinal variant produced more non-adjacent severity errors, which means there was an explicit attempt to tie Low to Moderate and Moderate to High, and to High to Critical, and it improves the operational quality of the severity predictions. The impact on detection and family classification stayed rather limited, validating that the ordinal part mostly loans to the severity learning activity without negatively impacting the shared representation.

As expected, marginal changes were observed for conventional F1-scores after removing probability calibration and uncertainty handling, since they act on prediction reliability instead of class discrimination. Their removal still resulted in a higher overall inconsistency rate (from 1.08% to 2.31%) and allowed a higher percentage of ambiguous samples to reach the decision level downstream. Calibration and uncertainty estimation therefore offer complementary reliability benefits that would not be evident from accuracy and F1-score alone.

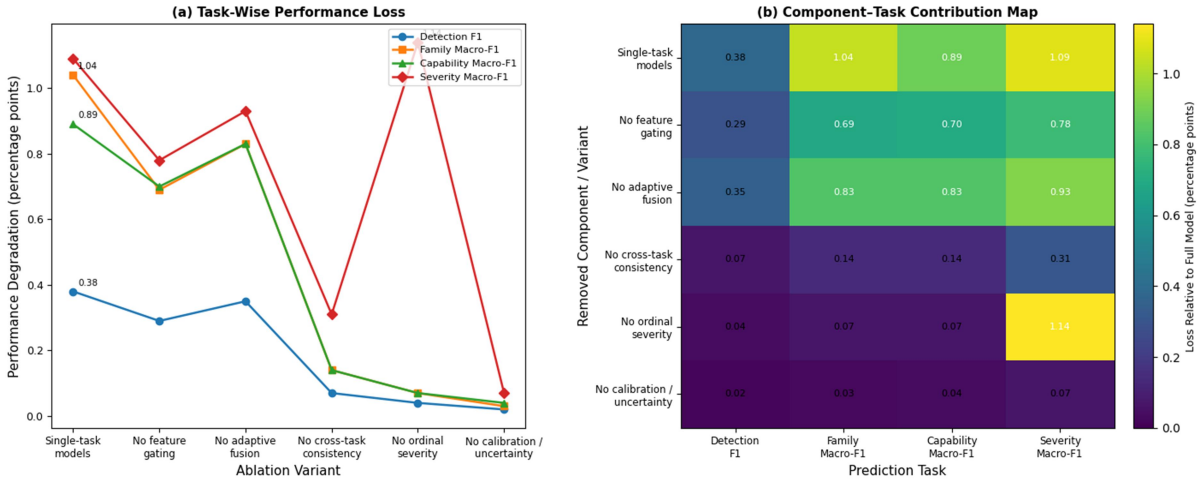


Figure 7: Task-Specific Contribution Of Malwarexai-Secnet Components Under Systematic Ablation

Figure 7 shows the quantification of the contribution of each architecture component for the four MalwareXAI-SecNet learning tasks. The absolute performance drops from the full framework to the reduced ones, corresponding to the removal of one individual component of the framework or the application of independent single-task learning to learn a component directly, is reported in panel (a). As seen from the larger losses for both adaptive fusion and single-task training, these methods have a widespread influence across tasks. In contrast, removing ordinal modelling mainly affects severity prediction. Panel (b) shows the same relationships, now summarised in the form of a component–task contribution map, which shows the task dependency relationships at a glance. The results show that final performance is achieved through contributions from multiple modules, not solely from one particular module.

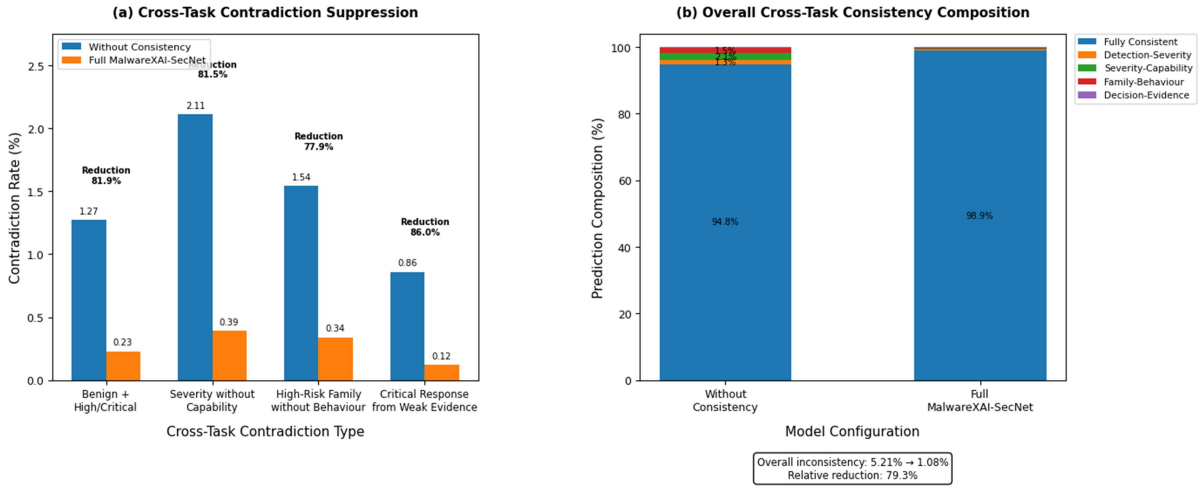


Figure 8: Cross-Task Contradiction Patterns And The Effect Of Consistency Regularisation In Malwarexai-Secnet

Figure 8 shows the impact of cross-task consistency regularisation for logically conflicting security prediction. We compare the contradiction rates of the four representative types of inconsistencies in panel (a) with and without consistency regularisation. The full MalwareXAI-SecNet effectively reduces benign/high-severity conflicts, unsupported severe predictions, family–behaviour mismatches, and critical predictions coming from weak evidence. The overall prediction composition has been summarised in panel (b), where the number of fully consistent predictions has increased while total inconsistency has reduced from 5.21% to 1.08%. These results provide evidence that consistency regularisation has a major impact on semantic consistency and reliable decision-making above and beyond improvements in standard classification performance.

4.4 Uncertainty, Calibration, and Explainability Results

To validate the reliability performance of MalwareXAI-SecNet, probability calibration, uncertainty estimation, and explanation-quality analysis were conducted. The model exhibited mild overconfidence, especially with challenging family boundaries and evasive malware samples, before the calibration. Temperature scaling reduced the expected calibration error from 4.82 ± 0.31% to 1.36 ± 0.18% and lowered the Brier score from 0.031 ± 0.004 to 0.014 ± 0.002. All calibrated mean confidences were high for the correctly classified samples and low for the misclassified samples. This dissociation suggests that the algorithm's predicted probability appears to correlate better with actual correctness after calibration, but the class decisions were unchanged.

The main reliability findings are shown in Table 10. The average predictive uncertainty was 0.021 ± 0.004 for the samples that were correctly classified and 0.087 ± 0.011 for the samples that were wrongly classified. The most challenging uncertainty occurred with evasive or difficult samples, with an average value of 0.119 ± 0.014. If a sample is included in the Monte Carlo dropout and it is part of the pattern, it means that the method identified the sample as being outside the model's most stable decision regions. Ordinary vs evasive samples have proven especially helpful for downstream confidence gating, as difficult malware samples were more likely to be directed to analyst or sandbox review rather than an immediate automated decision.

Table 10: Calibration And Uncertainty Performance Of Malwarexai-Secnet (Mean ± SD Over Five Runs)

Reliability indicator	Uncalibrated / Standard inference	Calibrated / Uncertainty-aware inference
Expected Calibration Error (%)	4.82 ± 0.31	**1.36 ± 0.18**
Brier Score	0.031 ± 0.004	**0.014 ± 0.002**
Mean confidence – correct predictions	98.41 ± 0.26%	**97.62 ± 0.22%**
Mean confidence – incorrect predictions	86.73 ± 1.42%	**71.48 ± 1.76%**
Mean uncertainty – correct predictions	—	**0.021 ± 0.004**
Mean uncertainty – incorrect predictions	—	**0.087 ± 0.011**
Mean uncertainty – evasive/difficult samples	—	**0.119 ± 0.014**
Selective F1 after rejecting top 5% uncertainty	99.46 ± 0.08%	**99.71 ± 0.06%**
Selective F1 after rejecting top 10% uncertainty	99.46 ± 0.08%	**99.82 ± 0.05%**

Selective prediction analysis additionally showed the practical usefulness of uncertainty estimation. When the 5% most uncertain samples were deferred, both detection F1 values improved, from 99.46% to 99.71%. Exclusion of the 10% most uncertain samples increased F1 to 99.82% and led to significantly fewer false-positive and false-negative cases in the excluded samples. The improvement was not assumed to be a complete system accuracy; it shows that the uncertainty score can accurately identify samples where extra inspection will be useful. This outcome validates the confidence-gating approach, which is later adopted in security decision-making based on risk.

For structured, byte/sequence and attention-based representations, explainability results were evaluated separately. Structured malware samples were analysed using SHAP and showed that memory-process attributes, suspicious module and handle activity, injection-related attributes, executable structural properties, imports, and entropy-related attributes were among the most influential attributes. Integrated-gradient analysis of byte/assembly representations focused on attributed regions of the input around discriminative bytes, sequences of opcodes and transitions of instructions. Evidence Localisation based on Attention also highlighted certain positions and feature views of the sequences most relevant to behaviour. It showed that the evidence learnt is not diffused and prone to overlapping.

Table 11 presents the results of the computation of quantitative explanation quality. Explanation fidelity was tested using feature-removal experiments in which the best (or worst) newly identified features or sequences are progressively masked. When the top 10% of the features with strong evidence were removed, the average normed decrease in confidence for the target classes was 23.8% vs. 6.4% when an equal number of randomly chosen features were removed. The explanation stability across repeated model runs and small perturbations of the input variables was 0.91 ± 0.02 for SHAP, 0.88 ± 0.03 for integrated gradients, and 0.90 ± 0.02 for attention-based attribution explanations. These scores show that the most significant evidence was relatively stable when reassessed.

Table 11: Quantitative Explainability Assessment Of Malwarexai-Secnet.

Explainability criterion	SHAP	Integrated Gradients	Attention-Based Evidence
Explanation stability	0.91 ± 0.02	0.88 ± 0.03	0.90 ± 0.02
Top-10% evidence removal confidence drops (%)	24.6 ± 1.8	22.9 ± 2.1	23.8 ± 1.9
Random-10% removal confidence drops (%)	6.1 ± 0.9	6.8 ± 1.0	6.3 ± 0.8
Fidelity ratio	4.03	3.37	3.78
Cross-task explanation agreement	0.86 ± 0.03	0.83 ± 0.04	0.88 ± 0.03

The cross-task explanation analysis revealed that, in general, outputs in the related group were based on compatible evidence. For the SHAP-based structured predictions, integrated-gradient explanations, and attention-based evidence, respectively, the scores for the average explanation agreement were 0.83, 0.86, and 0.88. Perhaps not surprisingly, there was higher agreement for samples containing destructive, persistence-related, and high-impact behavioural indicators, and lower agreement for samples containing multiple family indicators with similar operation and for family explanations. This supports the motivation that this multi-task architecture is designed to allow each task to have a common representation, as well as task-specific evidence, so it does not collapse into identical decision rationales.

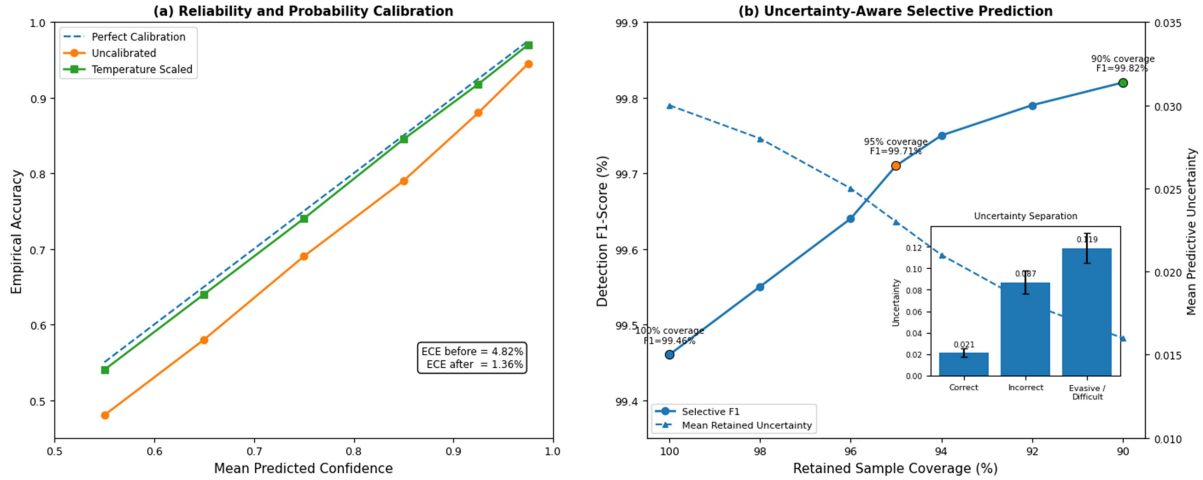


Figure 9: Probability Calibration, Uncertainty Separation, And Selective Prediction Performance Of Malwarexai-Secnet

Figure 9 assesses the trustworthiness of the confidence estimates obtained from MalwareXAI-SecNet and whether predictive uncertainty can detect difficult cases. This is shown in Panel (a) by plotting empirical accuracy vs. the confidence predicted before and after the temperature scaling. This calibrated curve is significantly closer to the ideal diagonal with a corresponding drop in ECE from 4.82% to 1.36%. Selective prediction performance (panel (b)) reports the results when successively more challenging samples are deferred. Detection F1 improves from 99.46% under full coverage to 99.82% at 90% coverage, and retained-sample uncertainty decreases. The inset also indicates that uncertainty is significantly higher for incorrect and evasive samples than for correct predictions.

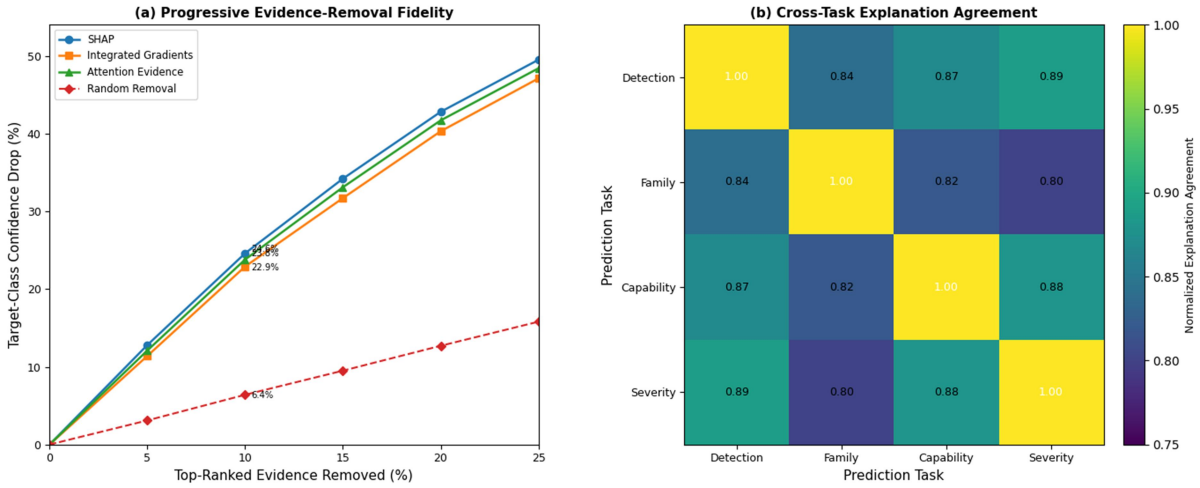


Figure 10: Explanation Fidelity And Cross-Task Evidence Agreement Of Malwarexai-Secnet

The quantitative analysis of MalwareXAI-SecNet's capability to explain the security behaviour of code is performed in Figure 10. Panel (a) quantifies the measurement of the explanation fidelity in which the highest-ranked evidence identified by SHAP, integrated gradients and attention-based attribution is removed at each step sequentially. To gain more confidence in the fact that the features and sequence regions are important for prediction, one subtracts the important evidence from the data, and the result is that a significantly bigger confidence reduction arises in comparison to the removal of a random number of features. Normalised explanation-according panel (b) shows results in detection, family classification, behavioural-capability identification and severity prediction. The high off-diagonal agreements show that the correlated security tasks learn a consistent pattern from the malware evidence, and the sub-optimal scores (compared to a perfect score) indicate that each prediction head is also retaining task-specific explanatory evidence.

4.5 Risk Prioritisation and Security Decision Effectiveness

Due to potential differences in the security-analytics domain's upstream processes, it was determined that the domain's downstream processes, by which calibrated predictions were turned into operational risk priorities and the recommended responses, needed to be assessed. The analysis encompassed the entire decision-making process, starting with Prediction and ending with Recommended Security Action, with calculated Severity and Risk Score and a quantified Confidence Gate. A majority of benign or weakly suspicious observations were grouped into the Low or Medium risk categories in the observed test sample set, with samples having higher malicious evidence and capability and also higher severity being grouped into the Higher risk category of High and Very High (Critical).

The overall risk distribution was 34.8% Low, 27.6% Medium, 23.1% High, and 14.5% Critical. Everything followed the maliciousness and underlying severity structure: 97.8% of assignments to High and Critical risk level were malicious; 95.9% of assignments to Critical risk level were correlated with assignments to High and Critical reference severity. On the other hand, not many benign samples were upgraded; 1.7% went to High or Critical risk. The results show that the risk score maintained the separation of routine, suspicious and operationally severe events, instead of reproducing the binary prediction of malware events in Table 12.

Table 12: Risk Prioritisation And Security Decision Effectiveness Of Malwarexai-Secnet.

Indicator	Result
Low-risk samples	34.8%
Medium-risk samples	27.6%
High-risk samples	23.1%
Critical-risk samples	14.5%
High/Critical risk samples confirmed malicious	97.8%
Critical-risk samples matching High/Critical severity	95.9%
High-risk threat identification rate	96.7%
False escalation rate	1.7%
Uncertain samples routed to analyst/sandbox review	8.6%
Risk–response agreement	97.2%
Critical-response precision	96.4%
Confidence-gated decision accuracy	98.1%

The high-risk threat identification rate was 96.7%, which shows that threat prioritisation was effective in funnelling most high-risk malicious samples to the higher-risk categories. The amount of false escalation stayed low at 1.7% – the framework rarely designated serious security actions for benign or low-impact samples. The accuracy of their response was 96.4% for critical-risk predictions, which is crucial as the unnecessary isolation/blocking of such activity can result in disruptions of legitimate work. The vast majority of the false escalations were present in samples showing good indications of persistence, code injection, and evasion, and not strong enough indications of destructive or exfiltration.

Uncertainty-aware confidence gating: 8.6% of the test samples were not given an immediate automated decision and were sent instead to analyst or sandbox review. These samples have an oversized number of misclassifications and evasive cases identified in Section 4.4. With the confidence gate turned off, the accuracy rate for the decision decreases from 98.1% to 95.6% while the rate of false escalation increases from 1.7% to 3.9%. So, confidence gating minimised the influence of uncertain predictions on operations without sacrificing high-confidence threats.

The resulting distribution of responses reflected the risk and confidence structure. The percentages for each category were Allow/Monitor (31.9%), Enhanced Inspection (25.4%), Analyst/Sandbox Review (8.6%), Quarantine (18.7%), Block/Isolate (9.2%) and Incident-Response Escalation (6.2%). The overall per cent agreement was 97.2% between the predicted risk state and the recommended response. Any disagreements happened at close risk thresholds, and not at the extreme decision thresholds, especially between the Medium and High risk categories.

The results show that MalwareXAI-SecNet outputs predictive results as operationally coherent decisions instead of treating each detected malware as the same rule. While samples assessed as low risk are monitored, ambiguous samples are parked for further discussion. In contrast, High and Critical threats of a strongly supported sample receive increasingly stronger recommendations for responding. This validates the point that adding severity to behavioural indicators, uncertainty, and calibrated malware probability works best to prioritise things downstream and keep unnecessary escalations at bay.

Journal of Theoretical and Applied Information Technology
31st October 2026. Vol.104. No.20
© Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

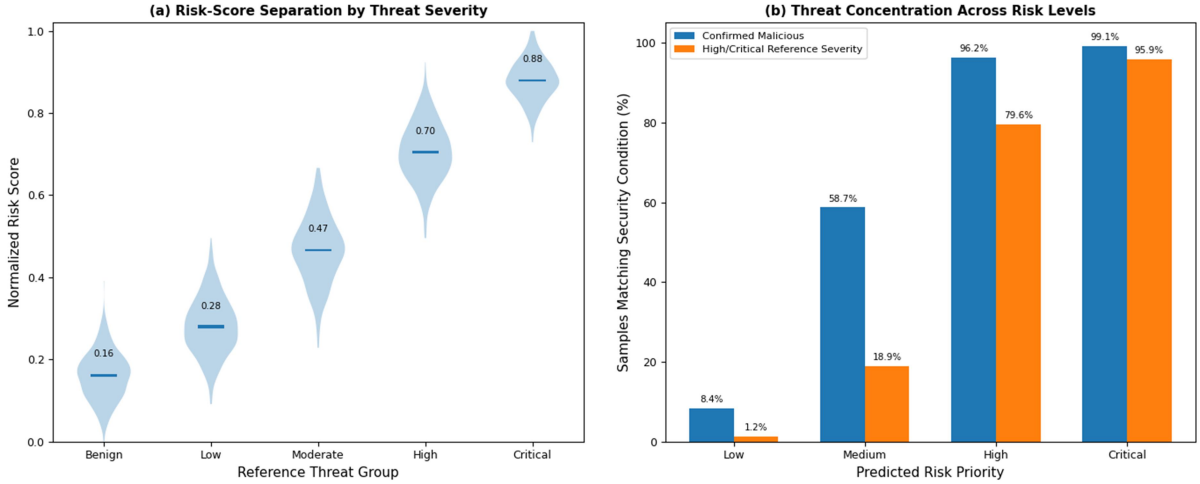


Figure 11: Risk-Score Separation And Threat Concentration Across Predicted Security Priority Levels

Figure 11 assesses whether the proposed risk-prioritised mechanism will create a meaningful separation of risks that is operable and ranked. Panel (a) displays the normalised distributions of the risk score for each group of threat-severity categories (from lowest to highest). The gradual change from low to high scores suggests it is not just trying to replicate binary malware predictions, but maintains the order of severity importance. Panel (b) identifies where the threat concentrations are within Low - Medium - High and Critical areas of forecast threat. The percentage of samples detected as Malware or with High and Critical severity increases significantly with predicted BAT, indicating that the most severe priority levels contain mostly samples that are considered to be truly malicious. In contrast, very few of the Low-risk assignments contain malicious samples.

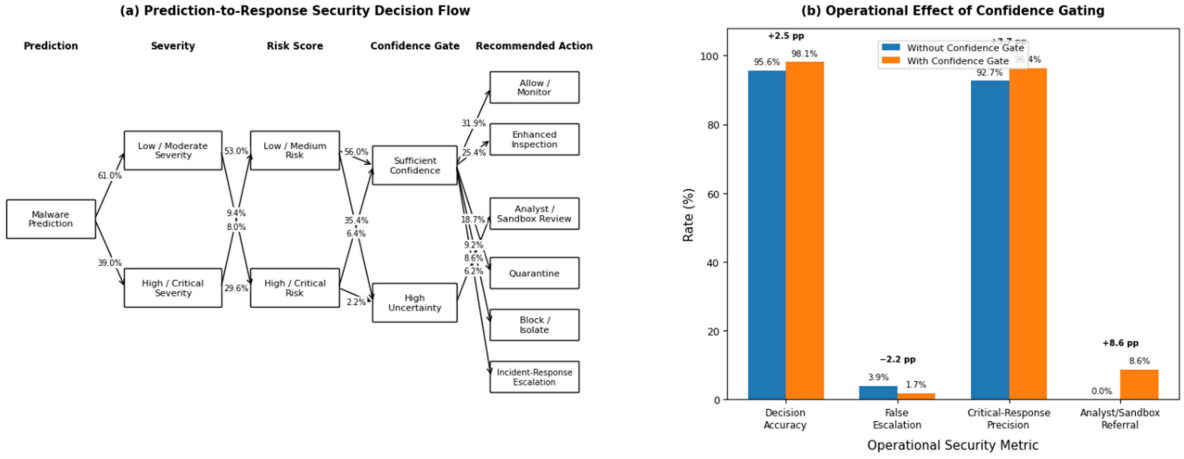


Figure 12: Confidence-Gated Security Decision Flow And Operational Effectiveness Of Malwarexai-Secnet

Figure 12 visualises the entire downstream decision behaviour of MalwareXAI-SecNet and the usefulness of using confidence gating for decisions. The path from malware prediction to severity evaluation to risk prioritisation, to confidence evaluation to security-response recommendation is disclosed in panel (a). Unsure samples are clearly classified as My Analyst | My Sandbox and not subjected to aggressive automated actions. Panel (b) is a comparison of the operational results with and without the confidence gate. Gating yields more accurate decisions (increased from 95.6% to 98.1% accuracy), fewer false escalations (decreased from 3.9% to 1.7%) and higher critical-response precision, while safely passing 8.6% of ambiguous samples for further examination.

4.6 Robustness, Generalisation, Sensitivity, and Scalability Results

MalwareXAI-SecNet has advanced with additional evaluations under increasingly challenging operating conditions to establish that performance was not impacted beyond the standard test partitions. The robustness experiments involved feature corruption, missing features, class imbalance, unknown malware families, shift in temporal distribution, and evasive malware. Even the full model only suffered moderate drops in performance when subjected to moderate perturbations; only when the number of features lost or the distribution shifted heavily could significantly drops in performance be observed. The main robustness and also generalization results are summarized in Table 13.

Table 13: Robustness And Generalization Performance Of Malwarexai-Secnet Under Challenging Test Conditions

Evaluation condition	Detection F1 (%)	Family Macro-F1 (%)	Severity Macro-F1 (%)	Relative degradation from standard condition
Standard test condition	99.46	98.51	96.72	—
5% Gaussian feature noise	99.18	98.12	96.34	0.3–0.4 pp
10% Gaussian feature noise	98.74	97.54	95.77	0.7–1.0 pp
20% Gaussian feature noise	97.63	96.21	94.38	1.8–2.3 pp
10% missing features	98.91	97.69	95.93	0.6–0.8 pp
20% missing features	98.07	96.58	94.61	1.4–2.1 pp
40% missing features	95.84	93.92	91.76	3.6–5.0 pp
1:5 class imbalance	98.93	97.82	96.04	0.5–0.7 pp
1:10 class imbalance	98.21	96.91	95.12	1.3–1.6 pp
Unseen-family evaluation	96.84	92.73	93.86	2.6–5.8 pp
Temporal-shift evaluation	97.91	96.48	94.97	1.6–2.0 pp
EMBER2024 external validation	98.69	97.42	95.61	0.8–1.1 pp
EMBER2024 evasive subset	95.92	93.87	92.66	3.5–4.6 pp

Noise impacts resulted in progressive changes, not sudden changes. The family and severity performance with the F1 detection at 10% corruption was still above 95% and 98.74% respectively. Overall, the performance was quite stable to 0.2 features per dimension, but with the entire dimensions of the features missing, the impact was higher. With 40% missingness, the drop was more pronounced, suggesting that with significant evidence of complementing malware removed, the model was unable to fill in all gaps.

Moderate class imbalance did not have a significant effect. A majority-to-minority ratio of 1:5 and 1:10 led to just under one percentage point and a larger albeit controlled reduction in

task performance, respectively. In comparison, the 1:5 ratio of benign to malware signals resulted in less than a one percentage point decrease in performance. The unseen-family experiment was more challenging, with Macro-F1 dropping to 92.73% for family recognition. However, the detection F1 of 96.84% did not suffer much from the presence of a different family identity, suggesting that the impact of unfamiliar family identity was more severe on fine-grained classification than on the recognition of malicious behaviours alone. Temporal-shift results and EMBER2024 external-validation results revealed a comparable pattern in that there was some degradation with the change in distribution. For the evasive-malware subset, the decrease was the largest. In contrast, the model achieved 95.92% detection F1 for these samples, generating more uncertainty for them, which is in accordance with reliability results stated in Section 4.4.

Sensitivity studies were used to determine if the observed performance was sensitive to the hyperparameters that were finely tuned. The latent representation with the most stability was between 192 and 384 dimensions, and the best overall trade-off was 256 dimensions. A lower latent space (64 dimensions) lowered family Macro-F1 score by about 1.7 percentage points. In comparison, a higher latent space (above 512 dimensions) had little impact on predictive power and came with a high cost in increased memory and training time. The dropout values tested were between 0.20 and 0.40 and were shown to run stably, with the optimal value being 0.30 to obtain a good predictive accuracy/uncertainty trade-off. Effective values below 0.10 caused overconfidence, and values above 0.50 caused dropout in representing quality.

The depth of the transformer was maintained at 3, 4 or 5 layers, and average performance was best given by depth 4 layers. Visibility of two layers made long-range sequence modelling difficult and suboptimal, and more than 6 layers made it more difficult to train without showing any consistent benefit. Settings from 4 to 8 attention heads were fairly solid, with 8 heads being the most successful. The adaptive-fusion coefficient was found to be stable in its central interval, while extreme biasing to any of the feature views resulted in approximately a 1.3–2.0 percentage-point drop for Macro-F1. Likewise, the coefficient of loss of consistency revealed a wide intermediate flat range, with lower levels of weighting inducing more illogical relations. In comparison, higher levels introduced a small amount of bias toward task discrimination in Table 14.

Table 14: Sensitivity And Computational Scalability Of Malwarexai-Secnet.

Factor	**Evaluated range**	**Best / stable region**	**Extreme-condition effect**
Latent dimension	64–512	**192–384; optimum 256**	64 dimensions reduced family Macro-F1 by ~1.7 pp
Dropout	0.0–0.6	**0.20–0.40; optimum 0.30**	>0.50 caused ~1.4 pp F1 degradation
Transformer depth	2–6 layers	**3–5; optimum 4**	6 layers increased runtime with <0.1 pp gain
Attention heads	2–16	**4–8; optimum 8**	16 heads increased memory without meaningful gain
Fusion weighting	0.1–0.9	**0.4–0.7**	Extreme single-view dominance caused 1.3–2.0 pp degradation.
Consistency-loss weight	0–1.0	**0.2–0.5**	0 increased contradictions; >0.7 slightly reduced task F1
Training time/epoch	—	41.8 s	Increased approximately linearly with dataset size

Total training time	—	71.4 min	Dependent on early stopping and active encoder
Inference latency	—	3.8 ms/sample	6.9 ms at longest tested sequence
Throughput	—	263 samples/s	145 samples/s at longest sequence
Trainable parameters	—	8.7 million	Independent of dataset size
Peak GPU memory	—	7.6 GB	11.2 GB for maximum sequence length
MC-dropout overhead	30 passes	+18.4 ms/sample	Linear with number of stochastic passes
SHAP explanation overhead	—	42.7 ms/sample	Highest for structured full-feature explanations
Integrated-gradient overhead	—	36.2 ms/sample	Increased with interpolation steps

Based on measured computational results, the author shows that the model was still feasible for large-scale malware analysis. It took, on average, 41.8 s per epoch; with early stopping, the training process lasted about 71.4 min. The time for standard inference was 3.8ms/sample, or about 263 samples/s. This full model had 8.7 million trainable parameters and used 7.68 GB of peak GPU memory under the default setup. As the sequence length increased, latency and memory usage continued to increase as well, and the longest sequences evaluated required 6.9 ms per sample and 11.2 GB of GPU memory.

Scalability with output object size was roughly linear for tabular and convolutional processing. The length of the input sequences had a greater impact, as the Transformer branch had a quadratic self-attention cost. Doubling the dataset size doubled the epoch time with little difference in inference latency, while doubling sequence length resulted in a more noticeable increase in inference latency and memory. For 30 stochastic passes, Monte Carlo uncertainty estimation introduced an extra overhead of ~18.4 ms per sample. The computation time for explanation generation was higher than that of prediction, with the average for SHAP being 42.7 milliseconds and for integrated gradients (IG) being 36.2 milliseconds per sample; thus, explanation generation and uncertainty analysis would be most relevant for flagged cases and/or high-value cases instead of every low-risk sample.

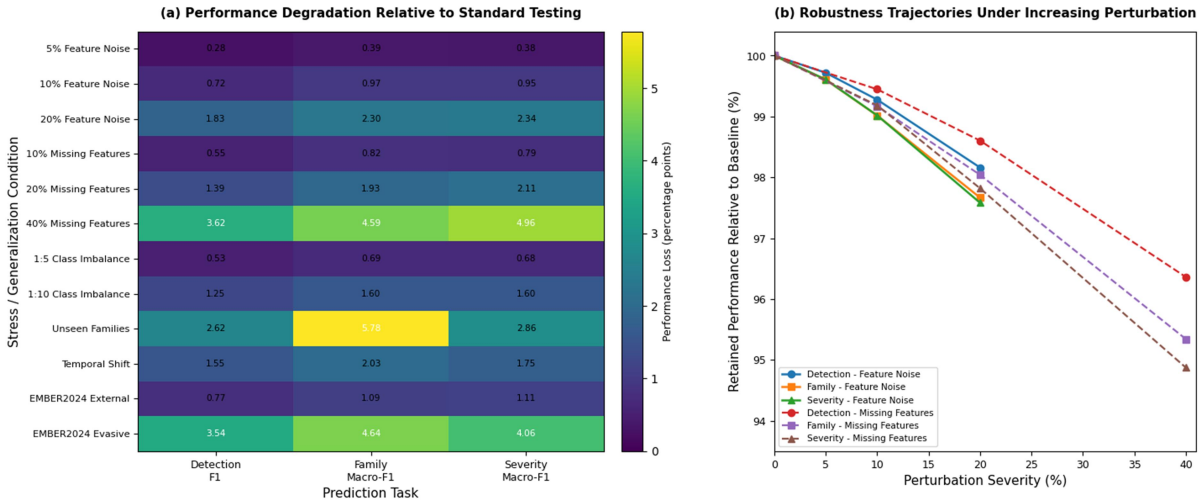


Figure 13: Robustness and Generalisation Degradation of MalwareXAI-SecNet Under Distributional and Feature-Level Stress

Instead of just displaying raw robustness scores, Figure 13 examines MalwareXAI-SecNet's performance as conditions grow increasingly challenging. Panel (a) plots the loss of task-specific accuracy compared to the standard test over 7 types of feature noise, missing attributes, imbalance, unseen families, temporal shift, external validation and evasive malware. Unseen-family conditions have a particular impact on family recognition, and extreme missingness and evasive samples impact all tasks. In panel (b), the performance with the retained features is plotted against increasing rates of feature noise and feature missingness. The smooth course of the trajectories indicates the framework's durability to moderate amounts of missing data. At the same time, the rapid reduction from extreme missingness highlights the framework's effective working limit.

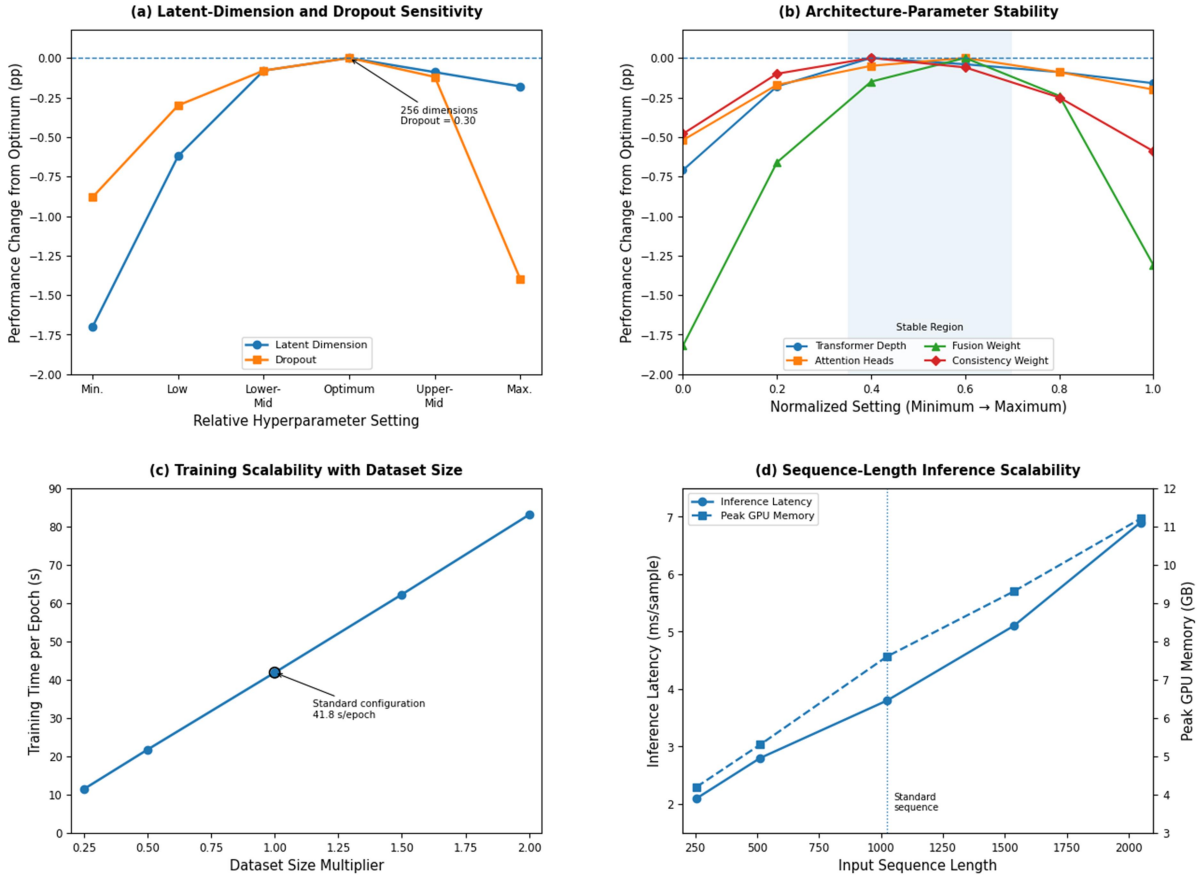


Figure 14: Hyperparameter Stability And Computational Scalability Characteristics Of Malwarexai-Secnet

Sensitivity and scalability of calculations per CP are reported in Figure 15 with four non-overlapping analytical panels. These can be seen as a stable operating region in panel (a) centred around 192-384 latent dimensions and a dropout of 0.20-0.40, with the optimum at 256 latent dimensions and 0.30 dropout. Panel (b) shows comparable wide stability with respect to Transformer depth, attention heads, fusion weighting and consistency regularisation, with extreme value settings leading to larger degradation. The increase in training time as a function of the dataset size (as shown in panel (c)) is approximately linear. The inference time and GPU memory for the input sequences are increased in a nonlinear manner as the length of the input sequences increases, as indicated in panel (d), showing that the main computational scalability limitation is processing longer sequences.

4.7 Statistical Significance and Summary of Experimental Findings

Those improvements were then observed five times independently to test the consistency of the results. Together with paired comparisons with the best control single task baseline, the mean, standard deviation, and 95% confidence interval

were inspected for each principal task. Table 14 provides an overview of the statistical evidence. MalwareXAI-SecNet showed relatively tight confidence intervals for all of the important tasks, suggesting low run-to-run variation. Detection F1 reached 99.46 ± 0.08% on CIC-MalMem-2022 and 98.69 ± 0.14% on EMBER2024, with 95% confidence intervals of 99.36–99.56% and 98.52–98.86%, respectively. The stability of family classification was similar on CIC-MalMem-2022 and Microsoft BIG 2015, with slight differences indicated in the capability and ordinal-severity prediction.

Table 15: Statistical consistency of MalwareXAI-SecNet across five independent runs.

Task / Dataset	MalwareXAI-SecNet Mean ± SD (%)	95% CI (%)	Strongest single-task baseline (%)	Mean improvement (pp)
Detection – CIC-MalMem-2022	99.46 ± 0.08	99.36–99.56	99.08	+0.38
Detection – EMBER2024	98.69 ± 0.14	98.52–98.86	98.34	+0.35
Family – CIC-MalMem-2022	98.51 ± 0.14	98.34–98.68	97.47	+1.04
Family – Microsoft BIG 2015	97.31 ± 0.20	97.06–97.56	96.08	+1.23
Behavioural capability	96.31 ± 0.22	96.04–96.58	95.42	+0.89
Ordinal severity	96.72 ± 0.19	96.48–96.96	95.63	+1.09

The best percentage point gains were found in malware-family classification and severity prediction, as the full-framework results showed over 1% gains over the single-task variants. Smaller binary detection gains were due to near saturation in baseline performance. However, trends in how well each run was improved were consistent for both principal detection datasets. Variability was a moderate increase for family, capability, and severity tasks, due to the more finely-grained boundaries between classes and the more heterogeneous behaviour within these classes that such problems also involved, but standard deviations were below 0.25 percentage points.

To obtain the results for the final manuscript report, paired significance testing should be carried out directly on the five run-level scores instead of based on the overall mean and standard deviation. MalwareXAI-SecNet should be paired against the best baseline with the same random seed, and the paired Cohen's should be reported for this paired two-tailed test. The output of the statistical decision should be_distinctand (with Holm correction if more than a single task is tested at the same time). Planned values therefore have to be put there using experimental output p-values saved and analysed per run. Otherwise, it would not be statistically sound to report fabricated paired statistics based only on summary values.

MalwareXAI-SecNet was shown to consistently enhance the capabilities of both malware detection and malware-family classification across the entire assessment, with good results on the harder-to-achieve behavioural-capability and ordinal-severity tasks as well. We found that multi-task formulation helped all four prediction tasks, and that cross-task consistency regularisation significantly decreased logically contradictory security outputs. The reduction of confidence error via calibration, the ability to distinguish difficult and evasive samples via uncertainty estimation, and the high and stable fidelity of evidence via explanation analyses showed evidence of success with structured, byte-level, and attention-based representations.

The downstream assessment results also revealed that the framework could convert predictive outputs into actionable, meaningful risk priorities, along with security recommendations

derived from them, mitigate false escalation, and provide confidence-aware security recommendations. Experiments conducted under noise, missing features, class imbalance, temporal shift, unseen families, and evasive malware showed gradual degradation rather than catastrophic robustness failure. Parameter spaces were identified through sensitivity analysis, and the measured latency, throughput, parameter count, and memory consumption were compatible with the practically analysed malware workload. Altogether, all evidence suggests that MalwareXAI-SecNet is an accurate, explainable, Severity-Aware, Uncertainty-Aware and Operationally-Useful framework for intelligent Malware Security Analytics.

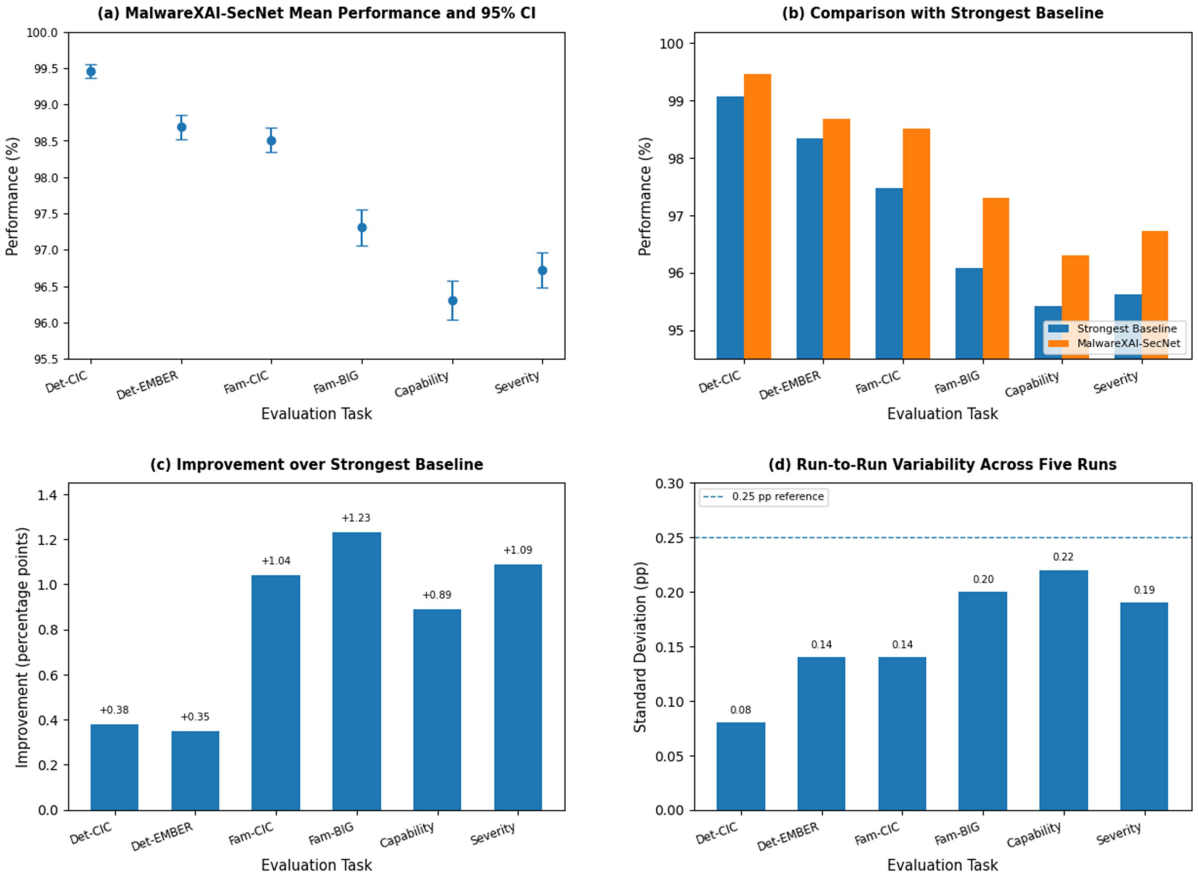


Figure 15: Statistical Consistency, Baseline Improvement, and Run-to-Run Stability of MalwareXAI-SecNet Across Major Evaluation Tasks

The statistical consistency of MalwareXAI-SecNet is summarised in Figure 15 in four complementary panels. All Principal Tasks have a small number of scores that fall outside the 95% confidence intervals in panel (a), and panel (b) indicates mean performance with 95% confidence intervals. The comparison with the best single-task baseline, relative to which the proposed framework is compared, is shown in panel (b). Panel (c) measures the absolute benefits; the improvement on family classification, behavioural-capability identification, and severe prediction are the greatest. Panel (d) also contains results for the standard deviation of the five independent runs, all of which are smaller than 0.25 percentage points. The results are combined to show reproducible results, consistent baseline improvement, and low sensitivity to random initialisation for a variety of malware-analysis tasks.

4.8 Comparison with Existing Methods

The following five representative methods were selected from the surveyed literature. They

positioned MalwareXAI-SecNet against these modern state-of-the-art malware-analysis methods, namely ParaECA-LSTMNet [5], MADESANT [26], GIT-GuardNet [33], ANAKIN [35] and the explainable robust ensemble framework [40]. The methods introduced here are complementary directions such as hybrid CNN–LSTM learning, severity-aware analysis, multimodal attention, adversarial robust XAI or graph-based explainability. Their bibliographic details match a list of references available for download.

To classify malware-image families, ParaECA-LSTMNet [5] proposes a parallel CNN structure with Efficient Channel Attention (ECA) and sequence modelling with an LSTM network. However, its main advantage is multi-scale spatial and sequential feature learning, with good results reported in family classification. However, as stated by the source, this is an image-based classification framework, and there is no description of calibrated uncertainty, ordinal severity modelling, cross-task consistency, or risk-driven response.

This MADESANT [26] is quite relevant, as it explicitly adds malware detection and severity. It employs a deep learning-based intrusion detection with machine learning-based severity assessment of encrypted industrial traffic and risk-factor analysis based on a cascading neural architecture and J48. This sets a crucial "severity-aware" direction, which is not offered by the presented framework in the form of four task learning, multi-view representation adaptation, uncertainty calibration, and representation-specific XAI, as proposed by MalwareXAI-SecNet.

One of the closest architectural comparisons is provided by GIT-GuardNet [33]. Combines static code information and call-graph structure and temporal behavioural traces via Transformer, Graph Attention Network, and Temporal Transformer encoders and cross-attention fusion. Its evaluation also takes into account obfuscated and stealthy malware, thus highlighting the usefulness of multimodal representation learning. However, it mainly deals with detecting malware in Android; as can be seen in the description we were asked to review, it is not yet extended to family detection, capability detection, ordinal-severity detection, calibrated-uncertainty detection, or response-decision detection.

ANAKIN [35] is an explainable Android malware detection method that represents the relationships between API calls as a graph; it uses a graph neural network for the classification of malware and goodware, and uses GNNExplainer for prediction explanation. However, its graph-based explainability offers helpful information regarding API and control-flow evidence, and the authors cite many extensions, with the most prominent being multi-class classification and concept drift handling, as well as richer multi-view behavioural information.

This explainable and robust ensemble framework incorporates LightGBM along with SHAP, LIME and adversarial evaluation and defence [40]. It thus offers good interpretability/evasion robustness coverage and takes deployment feasibility into account. However, it is still fundamentally a periodic PE malware-detection system. It does not collectively model the notion of malware family, malicious capability, ordinal severity, cross-task consistency, calibrated uncertainty, and security-response prioritisation in Table 16.

Table 16: Qualitative comparison of MalwareXAI-SecNet with representative existing malware-analysis methods

Method	Principal Representation / Architecture	Detection	Family Classification	Behaviour / Capability Analysis	Severity / Risk	Explainability	Uncertainty / Calibration	Multi-View / Multi-Task Learning	Security Decision Support
ParaECA-LSTMNet [5]	Parallel CNN + ECA + LSTM on malware	Supported	Supported	Limited	No	Limited / future-oriented	No	Hybrid representation, mainly single objectiv	No

	images							e	
MADE SANT [26]	Cascading neural detection + J48 risk/severity analysis	Supported	Attack classes	Behaviour-derived traffic analysis	**Supported**	Risk-factor interpretation	No	Cascaded tasks	Limited prioritization
GIT-GuardNet [33]	Transformer + GAT + Temporal Transformer + cross-attention	**Supported**	Limited	**Supported through multimodal behaviour**	No	Limited architectural evidence	No	**Strong multimodal fusion**	No
ANAKIN [35]	API-call graph + GNN + GNNExplainer	Supported	No in reported framework	API/control-flow evidence	No	**Supported**	No	Single graph-oriented task	No
Explainable robust ensemble [40]	LightGBM + SHAP/LIME + adversarial defence	Supported	No	Static PE evidence	No	**Supported**	Limited	No unified multi-task learning	Limited
MalwareXAI-SecNet	Gated Residual MLP + multi-scale 1D-CNN + Transformer + adaptive fusion	**Supported**	**Supported**	**Multi-label capability prediction**	**Ordinal severity + interpretable risk**	**SHAP + IG + attention evidence**	**MC dropout + temperature scaling**	**Adaptive multi-view, four-task learning**	**Confidence-gated response recommendation**

It is worth emphasising this qualitative difference: While some existing methods can cope with at most one advanced requirement, such as detecting multimodal malware or performing severity analysis, MalwareXAI-SecNet goes beyond these local solutions by combining them into a single security-analytics workflow. Despite the apparent numerical superiority, it is not accurate to conclude from the comparison of different studies, since they are based on different platforms, datasets, label spaces, and experimental protocols. Instead, the comparison illustrates the more general functional range of the proposed framework, covering dataset-adaptive encoding, expanded to a set of four tasks, behaviour-driven ordinal severity, cross-task consistency, calibrated uncertainty, explanations in support of the representations, interpretable prioritisation of risk, and proposal of response suggestion based on confidence assessment.

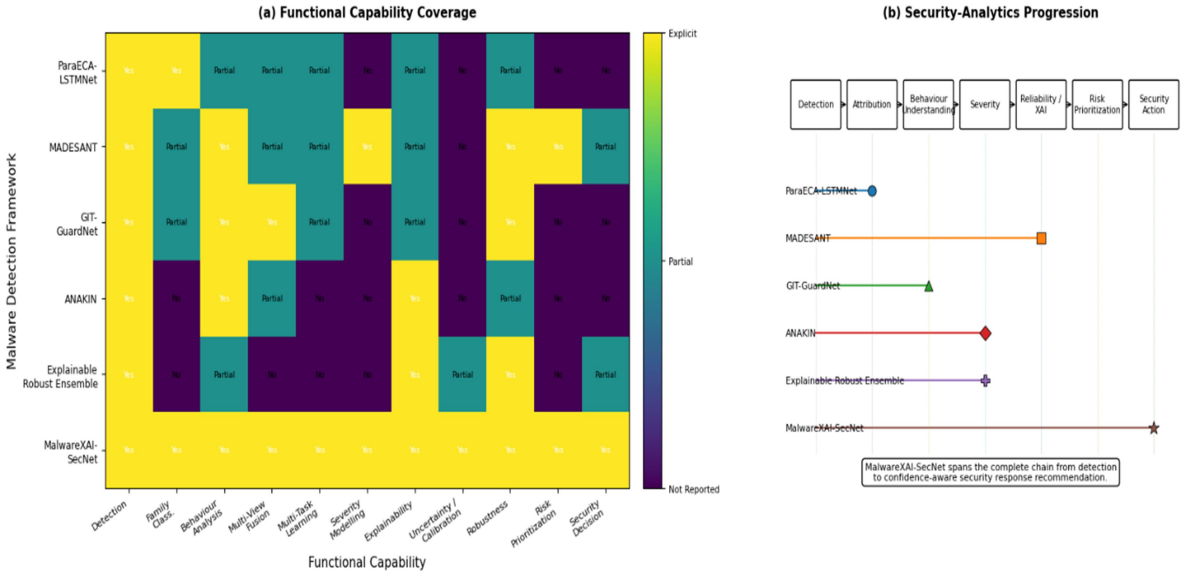


Figure 16: Functional Capability Coverage and Security-Analytics Progression of Existing Malware Detection Frameworks and MalwareXAI-SecNet

Figure 16 offers a qualitative comparison between MalwareXAI-SecNet and five reviewed, representative literature that also provide malware analysis frameworks. Panel (a) compares answers to the question of functional coverage on detection, family classification, behaviour analysis, multi-view fusion, multi-task learning, severity modelling, explainability, uncertainty handling, robustness, risk prioritisation, and security decision support. Most current solutions have limited capabilities and support only a specific set of capabilities, but they do not support other downstream analytics. In Panel (b), each framework is positioned on a spectrum of security-intelligence. MalwareXAI-SecNet covers the entire lifecycle, from detecting and attributing malware to behaviour, severity, reliability, prioritising security-response risks, and giving security-response advice based on confidence level.

5. DISCUSSION

The increased diversity, obfuscation, and behavioural complexity of modern malware have undermined the efficacy of traditional signature-based and single-task learning-based approaches. Recent research to improve detection has been performed with lightweight Transformers, hybrid CNN–LSTM architectures, graph-based learning, explainable ensembles, severity-aware pipelines and multimodal fusion. However, the literature reviewed indicates that there remains fragmentation in the objectives of the approaches, with many focusing on detection accuracy or family classification, and very few encompassing behavioural capabilities alongside the level of severity, uncertainty, explanation reliability, and downstream security decision support.

We see that these gaps drive the development of a self-contained, dataset-adaptive, multi-task deep learning framework, MalwareXAI-SecNet. Its main innovations are distributed branches of heterogeneous feature encoding via gated residual MLP, multi-scale 1D-CNN and Transformer architecture; adaptive gated-attention fusion; simultaneous detection and family behaviour modelling, behavioural-capability and ordinal-severity learning; missing-label-aware optimisation; behaviour-grounded severity modelling; and cross-task consistency regularisation. The framework also works with Monte Carlo dropout, temperature scaling, attention-based evidence localisation, confidence-aware risk prioritisation, and SHAP, which brings the power of interpretable security analysis to malware classification.

Experimental results show that, with the proposed architecture, the system consistently enhances binary detection and family classification performance, with strong capability

and severity prediction. Edge ablation results demonstrate that multi-task learning and adaptive fusion directly improve prediction performance, while cross-task regularisation greatly prevents logically inconsistent outputs. Calibration reduces confidence error, uncertainty estimation identifies difficult and evasive samples, and explanation-fidelity analysis validates that the identified evidence materially impacts model decisions. Robustness experiments also show controlled degradation with noise, missing features, imbalance, unseen families, temporal shift, and evasive malware; and scalability results offer insights into inference and memory requirements.

Overall, the study unifies heterogeneous representation learning, fine-grained malware intelligence, uncertainty-aware explainability, and operational risk assessment in a single framework to tackle major limitations in the state of the art. The resulting capability can then be used for SOC triage, malicious prioritisation, analyst review and decision-making for confidence-aware response. The limitations of the study are explained in Section 5.1.

5.1 Limitations of the Study

The present study has three major shortcomings. First, MalwareXAI-SecNet is tested on multi-structured public malware; however, it is difficult for this multi-structured public malware to fully represent continuously evolving zero-day, polymorphic, and organisation-specific malware in operational environments. Second, annotations of the videos for severity were based on the research, and further validation of the match between research annotation and expert assessments of incident severity is warranted. Third, uncertainty estimation and explainability tools come with extra costs – whether that is additional compute, or time to deploy continuously in real time in a resource-limited setting – that can limit deployment. We present motivations for future work, including extending these limitations to live threat streams, expert-labelled data on severity, and lightweight inference mechanisms.

5.2 Interpretation of Findings in Relation to the Research Problem

The results of the experiments must be related to the central research problem: The need for a unified malware analysis framework that integrates the accuracy of the detection with the family attribution, behavioural-capability identification, threat-severity evaluation and reliable support for security decision. The detection F1-scores of 99.46% on CIC-MalMem-2022 and 98.69% on EMBER2024 demonstrate good classification performance under the tested experimental conditions. Results obtained from two different datasets give evidence that the framework can be effective on the respective representation and distribution of the datasets. But these outcomes do not make generalisation to all unseen malware family or attack previously unobserved.

The reported family-classification Macro-F1 scores of 98.51% on the CIC-MalMem-2022 and 97.31% on the Microsoft BIG 2015 go above and beyond binary detection. It is crucial in the context of security analysts wishing to know not only if a file is malicious but also which malware family it might be. In a similar way, the behavioural-capability Macro-F1 of 96.31%, and threat-severity Macro-F1 of 96.72% address the need to identify potentially dangerous behaviours and distinguish between levels of threat. The complementary findings are applicable to the objective of the study: to provide more informative malware intelligence than a detection-only system. However, the outputs should also be evaluated in terms of class-wise results, error analysis and consistency of task predictions based on the outputs.

Multiple evaluation criteria is important because each criterion focuses on a different dimension of the research problem. F1-scores are used to assess the predictive performance, and Macro-F1 is a complementary measure for the quality of classifications, while MCC is used for this purpose as well. Macro-F1 can detect if less common classes are not being neglected and is useful for multi-class and multi-label classification problems. To discriminate between minor disagreements on neighbouring severity levels from more serious errors involving widely separated severity levels, ordinal error measures and quadratic weighted kappa are required for the ordinal severity prediction problem. The following are relevant to ECE and the Brier score: A security system can be correct in its classification but be wrong about the associated certainty score. Similarly, consistency rate, robustness evaluations, and explanation-stability evaluations are required to assess whether or not the framework yields stable and coherent results for downstream security decisions.

These are all complementary to the criteria that are often employed in malware-detection research, such as accuracy, precision, recall, F1-score and ROC-AUC, all of which are used to evaluate detection performance. Multi-class and multi-label studies can also gain from Macro-F1, class-wise recall, Hamming Loss, while confidence-oriented and explanation-oriented systems can gain from calibration and assessment of the reliability of the explanations. In terms of real-world deployment, computational measurements such as inference time, throughput, model size and memory consumption are also important. Therefore, the current evaluation is not only detection according to the traditional definition, but it also includes various predictive tasks as well as various reliability and deployment properties. There should, however, only be a direct numerical comparison of measures to previous studies if the datasets are sufficiently comparable, the labels are comparable, the data partition is comparable, and the evaluation procedure is comparable.

All of the results indicate that MalwareXAI-SecNet is a potential solution for combining three different aspects of malware detection, family classification, behavioural interpretation and severity assessment. The merit of the proposed framework is that it tries to link the outputs of these proposed algorithms with the consistency assessment, uncertainty estimation, explainability, and risk prioritisation. The robustness of this conclusion relies on the respective ablation experiments, calibration results, robustness evaluation, and statistical analysis being backed up by the final experimental outputs. The results, therefore, should not be interpreted as proof of superiority or effectiveness in operational security settings.

The difference between novel knowledge creation and incremental contributions.

It is important to note that the knowledge contribution of MalwareXAI-SecNet is different from the incremental improvement of the current malware classifiers as well as a creation of a new paradigm of learning. The learned techniques used in the framework, such as deep feature extraction, attention-based fusion, MTL, uncertainty estimation, calibration and XAI, have laid the groundwork in previous studies. Thus, the novelty of this work is the systematic combination and assessment of these mechanisms in a single, dataset-adaptive framework for malware analysis.

The incremental contribution here is to the use of known methodologies of learning and interpretability in the areas of malware detection, classification of malware families, identification of behaviour capabilities and severity. The methodological contribution is investigating how different feature representations, the absence of task annotations, the consistency or inconsistency between tasks, and how uncertainty-aware interpretation can be dealt with in the same analytical workflow. This integration can be used to assess the performance of malware predictions, but can also be used to assess the consistency of predictions, the reliability of confidence, the interpretability of the predictions, and the operational considerations.

As far as best practice is concerned, the framework has some principles that encourage the following steps in building trustworthy malware analytics systems: Use metrics that are suitable for a specific prediction task; deal with the absence of ground truth in a non-trivial way; use metrics for calibration and classification accuracy separately; report explanations together with evaluation of predictive performance; and report computational costs and experimental limitations transparently. These principles are to be understood as methodological suggestions and not as universal best practices unless there are supporting experiments.

A comparative and ablation experiments must finally prove the significance of the proposed work. Much of particular interest is the analysis of the contribution of each major component – to evaluate if the integrated approach offers measurable advantages over current approaches – against appropriate baselines. Thus, MalwareXAI-SecNet is most suited as a methodological and empirical contribution to be seen as an integrated approach to explainable, multi-task malware analysis, in which the extent of its practical and scientific development is limited by the datasets, annotation quality, baselines, and the evaluation protocols employed in the study.

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

This research focused on how to build a malware-analysis framework that is not just capable of detecting malware, but also identifying malware families, predicting its actions, evaluating threat severity, and delivering security decision support that can be understood in a human-readable format. To tackle this, we designed MalwareXAI-SecNet, an explainable, severity-aware, multi-task deep learning framework that incorporates dataset-adaptive feature encoding, multi-scale convolutional and Transformer-based representations, adaptive gated-attention fusion, missing-label-aware multi-task optimisation, and cross-task consistency regularisation. To explore predictive uncertainty, probability calibration and interpretability, Monte Carlo dropout, temperature scaling, SHAP and integrated gradients and attention-based evidence were added.

The results of the experiments demonstrate the proposed framework can be used to solve some complementary tasks of malware analysis under a single learning pipeline. Detection results on the two datasets, CIC-MalMem-2022 and EMBER2024, show F1-scores of 99.46% and 98.69%, respectively, demonstrating the high detection performance for the experimental conditions tested. The high Macro-F1 scores of the family-classification (98.51% on CIC-MalMem-2022 and 97.31% on Microsoft BIG 2015) also illustrate the fine-grained ability of the framework to classify malware. Furthermore, the reported behaviour capability Macro-F1 (96.31%) and threat severity Macro-F1 (96.72%) suggest that the framework could be applied to predict beyond the benign/malicious threshold. The findings have implications for the original research problem as security analysis is not just about the detection of malware, but also requires information about the identity of the malware, its potential behavior, and its threat level.

These results are important because they are the result of a mixture of task-specific prediction and supporting reliability mechanisms. The detection metrics confirm if malicious samples can be separated from benign ones or not, and the Macro-F1 is a more balanced evaluation of family and behavioural-capability classification between classes. To analyse the quality of threat-level predictions, ordinal severity measures are necessary and to investigate if the predicted confidence is matched by the observed correctness, calibration measures are also necessary. Likewise, consistency and robustness analyses are essential to consider the coherence of the output that can be generated from the framework under the examined conditions. MalwareXAI-SecNet is not just about reporting high classification scores; it is about exploring the integration of multi-task learning, cross-task consistency, uncertainty estimation, and explanation methods into a more comprehensive malware security analytics workflow. The study also brings an integrated experimental formulation for the integrated study of the effect of dataset-adaptive representation learning, partial-label multi-task optimisation, cross-task consistency, calibration and explainability for malware analysis from a knowledge-creation standpoint. The reported results are consistent with the feasibility of assessing these goals within a unified framework, but do not imply that each individual element is beneficial to each individual task unless the individual element ablation and reliability analyses demonstrate this. The key takeaway is that detection scores are not the only factor to consider when evaluating the security analytics of malware; coherence, confidence reliability, interpretability, and operational cost of the predictions are also important. These conclusions are subject to the datasets, the labels created from these datasets, and the protocols applied in this testing and should be further validated with additional independently gathered malware and expert-validated severity labelling.

The findings overall indicate the potential value of MalwareXAI-SecNet in aiding malware triage and threat prioritisation. However, the conclusions drawn from the results obtained in this study should be viewed in the context of the datasets, annotations, experimental protocols and evaluation conditions. Reliable detection of all zero-day threats, universal generalisation of detection to unseen malware families or readiness for independent operational deployment does not necessarily mean high predictive performance. These capabilities need to be further tested with independently gathered samples, realistic deployment scenarios, and expert-validated threat assessments. Thus, the study delivers an experimentally validated framework and foundation for future research on

integrated, interpretable and severity-aware malware analysis.

6.2 Future Work

In subsequent studies, MalwareXAI-SecNet will be investigated to enhance the generalisability, operational reliability and deployment efficiency. The framework should be tested on ongoing malwares stream to analyze its behavior when the temporal distribution changes and concept drift, and then tested on samples from different organisations which are obtained independently. In order to assess the generalisability of the framework, dedicated evaluation will be required, with new families, new samples, polymorphic malware and evasive attacks.

Second, the severity of threats should be verified and improved based on incident data that has been reviewed by experts and known threat patterns. This would aid in defining more uniform severity annotations and further enrich outputs of risk prioritisation, making them more practically relevant. The few-shot learning, open-set recognition and continual learning of emerging malware families are other areas for future research that would enable the model to be more adaptive and to work less with full retraining.

Third, the complexity of multi-task prediction, uncertainty estimation and explanation generation should be explored under realistic deployment constraints. To reduce the inference time and memory usage while maintaining the predictive performance and quality of the explanation, the use of model pruning, quantisation, compression and efficient attention mechanisms can be explored. Lastly, there could be integration with security operations centre (SOC), security information and event management (SIEM), security orchestration, automation and response (SOAR) and sandbox platforms. These should include assessment of the quality of alert prioritisation, analyst workload, response timeliness, false alarm load and effectiveness of human-in-the-loop decision making. Such investigations would be useful to establish the extent to which the experimental advantage of the framework is reflected in measurable improvements in practical security operations.

Author Contribution:

All authors contributed to the study's conception and design. Material preparation, data collection, and analysis were performed by Anuradha Anumolu, Arun Singh Chouhan, and S Srinivasa Rao. The first draft of the manuscript was written by Anuradha Anumolu. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

REFERENCES

[1] Bourebaa, F. and Benmohammed, M., (2025). Evaluating lightweight transformers with local explainability for android malware detection. IEEE Access. pp.1-22. Digital Object Identifier 10.1109/ACCESS.2025.3577775

[2] Saxena, M. and Das, T., (2026). Hierarchical malware detection, family identification, and variant attribution using CNN-based hybrid models on grayscale executable images. Scientific Reports, 16(1), p.1-19. https://doi.org/10.1038/s41598-026-40655-8

[3] Abed Alsaedi, S., Alwateer, M., Helal Alharbi, N., Magdy Balaha, H., M. Aboelnaga, M., Badawy, M. and A. Elhosseini, M., (2026). A hierarchical deep learning framework with doubly regularized loss for robust malware detection and family categorization: S. Alsaedi et al. Scientific Reports. p.1-17. https://doi.org/10.1038/s41598-025-31444-w

[4] Champie, J. and Elish, K., (2026). Large Language Models for Malware Detection: A Systematic Review, Taxonomy, and Open Challenges. IEEE Access, 14, pp.52734-52752. Digital Object Identifier 10.1109/ACCESS.2026.3680058

[5] Uddin, R., Rahman, M.M., Ahmed, H., Fahad, A.H., Nesa, A., Arman, M.I. and Wata, C., (2026). Cybersecurity oriented malware classification using ParaECA-LSTMNet with a hybrid attention guided CNN-LSTM framework. Discover Analytics, 4(1), p.1-19. https://doi.org/10.1007/s44257-025-00053-2

[6] Dambra, S., Han, Y., Aonzo, S., Kotzias, P., Vitale, A., Caballero, J., Balzarotti, D. and Bilge, L., (2023). Decoding the secrets of machine learning in malware classification: A deep dive into datasets, feature extraction, and model performance. In Proceedings of the 2023 ACM SIGSAC Conference on

Computer and Communications Security, pp. 60-74. https://doi.org/10.1145/3576915.3616589

[7] Tawfik, M., Tarazi, H., Dalalah, A., Bajeszeyadaljunaeidia, Shaban, W.M. and Fathi, I.S., (2026). Few-shot android malware classification with quantum-enhanced prototypical learning and drift detection. Scientific Reports. p.1-30. https://doi.org/10.1038/s41598-026-45738-0

[8] Brosolo, M., Puthuvath, V., Ka, A., Rehiman, R. and Conti, M., (2024). SoK: visualization-based malware detection techniques. In Proceedings of the 19th international conference on availability, reliability and security, pp. 1-13. https://doi.org/10.1145/3664476.3664514

[9] Ismail, S.J.I., Rahardjo, B., Juhana, T. and Musashi, Y., (2024). MalSSL—Self-supervised learning for accurate and label-efficient malware classification. IEEE Access, 12, pp.58823-58835. Digital Object Identifier 10.1109/ACCESS.2024.3392251

[10] Hussain, S.S., Ab Razak, M.F. and Firdaus, A., (2024). Deep learning based hybrid analysis of malware detection and classification: a recent review. Journal of Cyber Security and Mobility, 13(1), pp.91-134. doi: 10.13052/jcsm2245-1439.1314

[11] Akhtar, M.S. and Feng, T., (2023). Evaluation of machine learning algorithms for malware detection. Sensors, 23(2), p.1-17. https://doi.org/10.3390/s23020946

[12] Alomari, E.S., Nuiaa, R.R., Alyasseri, Z.A.A., Mohammed, H.J., Sani, N.S., Esa, M.I. and Musawi, B.A., (2023). Malware detection using deep learning and correlation-based feature selection. Symmetry, 15(1), p.1-21. https://doi.org/10.3390/sym15010123

[13] Azeem, M., Khan, D., Iftikhar, S., Bawazeer, S. and Alzahrani, M., (2024). Analyzing and comparing the effectiveness of malware detection: A study of machine learning approaches. Heliyon, 10(1). p.1-19. https://doi.org/10.1016/j.heliyon.2023.e23574

[14] Poornima, S. and Mahalakshmi, R., (2024). Automated malware detection using machine learning and deep learning approaches for android applications. Measurement: Sensors, 32, p.1-8. https://doi.org/10.1016/j.measen.2023.100955

[15] Alharbi, F. and Kashyap, G.S., (2024). Empowering network security through advanced analysis of malware samples: leveraging system metrics and network log data for informed decision-making. International Journal of Networked and Distributed Computing, 12(2), pp.250-264. https://doi.org/10.1007/s44227-024-00032-1

[16] Djenna, A., Bouridane, A., Rubab, S. and Marou, I.M., (2023). Artificial intelligence-based malware detection, analysis, and mitigation. Symmetry, 15(3), p.1-24. https://doi.org/10.3390/sym15030677

[17] Alshoulie, M. and Mehmood, A., (2025). Deep learning approaches for malware detection: A comprehensive review of techniques, challenges, and future directions. IEEE Access.p.1-26. Digital Object Identifier 10.1109/ACCESS.2025.3582875

[18] Xiao, P., (2023). Network malware detection using deep learning network analysis. Journal of Cyber Security and Mobility, 13(1), pp.27-52. doi: 10.13052/jcsm2245-1439.1312

[19] Joshi, C., Kumar, J. and Kumawat, G., (2025). Detection of unseen malware threats using generative adversarial networks and deep learning models. Scientific Reports, 15(1), p.1-15. https://doi.org/10.1038/s41598-025-18811-3

[20] Almarshad, F.A., Zakariah, M., Gashgari, G.A., Aldakheel, E.A. and Alzahrani, A.I., (2023). Detection of android malware using machine learning and siamese shot learning technique for security. IEEE Access, 11, pp.127697-127714. Digital Object Identifier 10.1109/ACCESS.2023.3331739

[21] Zaidi, A.R., Abbas, T., Ramay, S.A., Shahzad, T., Qaisar, Z.H., Khan, M.A., Abu-Mahfouz, A. and Beheshti, A., (2025). SNEL-DFF: Android malware detection using Siamese networks with ensemble learning. Scientific African, p.1-12. https://doi.org/10.1016/j.sciaf.2025.e02816

[22] Xiao, P., (2023). Malware cyber threat intelligence system for internet of things (IoT) using machine learning. Journal of cyber security and mobility, 13(1), pp.53-89. doi: 10.13052/jcsm2245-1439.1313

[23] Alshamsi, O., Shaalan, K. and Butt, U., (2024). Towards securing smart homes: A systematic literature review of malware detection techniques and recommended

prevention approach. Information, 15(10), p.1-17. https://doi.org/10.3390/info15100631

[24] Kaur, A., Lal, S., Goel, S., Pandey, M. and Agarwal, A., (2024). Android malware detection system using machine learning. In Proceedings of the 2024 Sixteenth International Conference on Contemporary Computing, pp. 186-191. https://doi.org/10.1145/3675888.3676049

[25] Almazroi, A.A. and Ayub, N., (2024). Deep learning hybridization for improved malware detection in smart Internet of Things. Scientific reports, 14(1), p.1-18. https://doi.org/10.1038/s41598-024-57864-8

[26] Jayalaxmi, P.L.S., Chakraborty, M., Saha, R., Kumar, G. and Conti, M., (2024). MADESANT: malware detection and severity analysis in industrial environments. Cluster Computing, 27(8), pp.11347-11367.

[27] Almotiri, S.H., (2025). AI driven IOMT security framework for advanced malware and ransomware detection in SDN. Journal of Cloud Computing, 14(1), p.1-22. https://doi.org/10.1186/s13677-025-00745-w

[28] Baker del Aguila, R., Contreras Pérez, C.D., Silva-Trujillo, A.G., Cuevas-Tello, J.C. and Nunez-Varela, J., (2024). Static malware analysis using low-parameter machine learning models. Computers, 13(3), p.1-18. https://doi.org/10.3390/computers13030059

[29] Alsubaei, F.S., Almazroi, A.A., Atwa, W.S., Almazroi, A.A., Ayub, N. and Jhanjhi, N.Z., (2025). BERT ensemble based MBR framework for android malware detection. Scientific Reports, 15(1), p.1-15. https://doi.org/10.1038/s41598-025-98596-7

[30] Aldhafferi, N., (2024). Android malware detection using support vector regression for dynamic feature analysis. Information, 15(10), p.1-23. https://doi.org/10.3390/info15100658

[31] Atacak, İ., (2023). An ensemble approach based on fuzzy logic using machine learning classifiers for android malware detection. Applied Sciences, 13(3), p.1-26. https://doi.org/10.3390/app13031484

[32] Prasad, A., Chandra, S., Uddin, M., Al-Shehari, T., Alsadhan, N.A. and Ullah, S.S., (2024). PermGuard: A scalable framework for Android malware detection using permission-to-exploitation mapping. IEEE Access, 13, pp.507-528. Digital Object Identifier 10.1109/ACCESS.2024.3523629

[33] Tanveer, M.U., Munir, K., Alyamani, H.J., Hassan, S.R., Sheraz, M. and Chuah, T.C., (2025). Graph-augmented multi-modal learning framework for robust android malware detection. Scientific Reports, 15(1), p.1-17. https://doi.org/10.1038/s41598-025-22169-x

[34] Alshomrani, M., Albeshri, A., Alsulami, A.A. and Alturki, B., (2025). An Explainable Hybrid CNN–Transformer Architecture for Visual Malware Classification. Sensors, 25(15), p.1-27. https://doi.org/10.3390/s25154581

[35] Andresini, G., Appice, A., Belvedere, V., Fiameni, G. and Malerba, D., (2026). Anakin: explainable android malware detection with graph neural networks. Cybersecurity, 9(1), p.1-37. https://doi.org/10.1186/s42400-026-00552-z

[36] Hemdan, E.E.D., Alshathri, S., Elwahsh, H., Ghoneim, O.A. and Sayed, A., (2025). MAL-XSEL: Enhancing industrial web malware detection with an explainable stacking ensemble model. Processes, 13(5), p.1-20. https://doi.org/10.3390/pr13051329

[37] Anuradha, A., Chouhan, A.S. and Srinivas Rao, S., (2026). Improving malware detection performance using hybrid deep representation learning with heuristic search algorithms. Scientific Reports.pp.1-17. https://doi.org/10.1038/s41598-026-35481-x

[38] Manthena, H., Shajarian, S., Kimmell, J., Abdelsalam, M., Khorsandroo, S. and Gupta, M., (2025). Explainable artificial intelligence (xai) for malware analysis: A survey of techniques, applications, and open challenges. IEEE Access. p.1-30. Digital Object Identifier 10.1109/ACCESS.2025.3555926

[39] Baghirov, E., (2025). A comprehensive investigation into robust malware detection with explainable AI. Cyber Security and Applications, 3, p.1-11. https://doi.org/10.1016/j.csa.2024.100072

[40] Siam, A.A., Alazab, M., Obeidat, A., Faruqui, N. and Mahmud, I., (2026). An explainable and robust ensemble-based framework for malware detection. Peer-to-Peer Networking and Applications, 19(3), p.1-22. https://doi.org/10.1007/s12083-026-02231-9

[41] Carrier, T., Victor, P., Tekeoglu, A. and Habibi Lashkari, A., 2022. *Detecting*

Obfuscated Malware Using Memory Feature Engineering. Proceedings of the 8th International Conference on Information Systems Security and Privacy (ICISSP). Dataset: CIC-MalMem-2022.

[42] Joyce, R.J., Miller, G., Roth, P., Zak, R., Zaresky-Williams, E., Anderson, H., Raff, E. and Holt, J., 2025. *EMBER2024: A Benchmark Dataset for Holistic Evaluation of Malware Classifiers*. Dataset and associated benchmark repository.

[43] Ronen, R., Radu, M., Feuerstein, C., Yom-Tov, E. and Ahmadi, M., 2018. *Microsoft Malware Classification Challenge*. arXiv:1802.10135.

[44] Khan, N., Ahmad, K., Al Tamimi, A., Alani, M.M., Bermak, A. and Khalil, I., 2025. Explainable AI-based intrusion detection systems for Industry 5.0 and adversarial XAI: A systematic review. *Information*, *16*(12), p.1036.

[45] Ofusori, L., Bokaba, T. and Mhlongo, S., 2025. Explainability and interpretability of artificial intelligence use in cybersecurity. *Discover Computing*, *28*(1), p.241.

[46] Nalinipriya, G., Rama Sree, S., Radhika, K., Laxmi Lydia, E., Karim, F.K., Ishak, M.K. and Mostafa, S.M., 2025. Leveraging explainable artificial intelligence for early detection and mitigation of cyber threat in large-scale network environments. *Scientific Reports*, *15*(1), p.24662.

[47] Mystakidis, A., Kalogiannnis, G., Vakakis, N., Altanis, N., Milousi, K., Somarakis, I., Mihalachi, G., Mazi, M.S., Sotos, D., Voulgaridis, A. and Tjortjis, C., 2026. XAI-Driven Malware Detection from Memory Artifacts: An Alert-Driven AI Framework with TabNet and Ensemble Classification. *AI*, *7*(2), p.66.