



# OPEN Multimodal malware classification using proposed ensemble deep neural network framework

Sadia Nazim<sup>1,6,7</sup>, Muhammad Mansoor Alam<sup>2,7</sup>, Safdar Rizvi<sup>3,7</sup>, Jawahir Che Mustapha<sup>1,7</sup>✉, Syed Shujaa Hussain<sup>4,7</sup> & Mazliham Mohd Su'ud<sup>5,7</sup>

In the contemporary technological world, fortifying cybersecurity defense against dynamic threat landscapes is imperative. Malware detectors play a critical role in this endeavor, utilizing various techniques such as statistical analysis, static and dynamic analysis, and machine learning (ML) to compare signatures and identify threats. Deep learning (DL) aids in accurately classifying complex malware features. The cross-domain research in data fusion strives to integrate information from multiple sources to augment reliability and minimize errors in detecting sophisticated cyber threats. This collaborative approach is the least addressed and pivotal for protecting against the advancing environment of modern malware attacks. This study presents a state-of-the-art malware analysis framework that employs a multimodal approach by integrating malware images and numeric features for effective malware classification. The experiments are performed sequentially, encompassing data preprocessing, feature selection using Neighbourhood Component Analysis (NCA), and dataset balancing with Synthetic Minority Over-sampling Technique (SMOTE). Subsequently, the late fusion technique is utilized for multimodal classification by employing Random Under Sampling and Boosting (RUSBoost) and the proposed ensemble deep neural network. The RUSBoost technique involves random undersampling and adaptive boosting to moderate bias toward majority classes while improving minority class (malware) detection. Multimodal Late fusion experimental results (95.36%) of RUSBoost (numeric) and the proposed model (imagery) outperform the standalone prevailing results for imagery (95.02%) and numeric (93.36%) data. The effectiveness of the proposed model is verified through the evaluation metrics such as Recall (86.5%), F1-score (85.0%), and Precision (79.9%). The multimodal late fusion of numeric and visual data makes the model more robust in detecting diverse malware variants. The experimental outcomes demonstrate that multimodal analysis may efficiently increase the identification strength and accuracy, especially when majority vote and bagging are employed for late fusion.

**Keywords** Malware detection, Multimodal analysis, Deep learning, Late fusion, Android malware, Artificial intelligence, Cyberattacks

The proliferation of the World Wide Web and the advances in digital technology have brought about a remarkable transformation for people to accomplish their tasks. Technological developments have offered individuals the ability to organize a variety of events online. These events involve various activities, including communicating with friends, handling capital transactions, and observing physical well-being. These modernizations have transformed how people collaborate with the world, making everyday responsibilities more effective and approachable. However, these technological advances have also allowed cyberattackers to cause damage remotely. According to recent research conducted by industry experts, cyberattacks have resulted in trillions of dollars in financial losses to the global economy. Among the most commonly used tools by cyber attackers to initiate these attacks is malware. Essentially, any software that performs unauthorized or suspicious activities on a target

<sup>1</sup>Malaysian Institute of Information Technology, Universiti Kuala Lumpur, 1016, Jalan Sultan Ismail, 50250 Kuala Lumpur, Malaysia. <sup>2</sup>Faculty of Computing, Riphah International University, Near Hajj Complex I-14, Islamabad, Pakistan. <sup>3</sup>Department of Computer Science, Bahria University Karachi Campus, 13 National Stadium Rd, Karachi, Pakistan. <sup>4</sup>Department of Computer Science, Sir Syed CASE Institute of Technology, Street 33, Block A Sector B-17 Multi Gardens B-17, Islamabad, Pakistan. <sup>5</sup>Multimedia University, Persiaran Multimedia, Cyberjaya, Selangor, Malaysia. <sup>6</sup>Department of Computer Science, Bahria University Islamabad Campus, Shangrilla Rd, E-8/1, Islamabad, Pakistan. <sup>7</sup>Sadia Nazim, Muhammad Mansoor Alam, Safdar Rizvi, Jawahir Che Mustapha, Syed Shujaa Hussain and Mazliham Mohd Su'ud: These authors contributed equally to this work. ✉email: jawahir@unikl.edu.my

computer is classified as malware<sup>1</sup>. Protecting your computing devices from malware is crucial to protecting sensitive personal data. It would be feasible to take suitable precautions by being aware of the various types of malicious software, including viruses, trojan horses, worms, and ransomware. Even novel malware categories that use innovative methods like packing and encryption are blocked with the appropriate action. To protect e-activities, it is imperative to be acquainted and careful<sup>2</sup>.

The exploration of malware analysis is a beneficial procedure for the consideration of internal activities and the performance of malicious software. This process utilizes diverse approaches designed to examine malware and secure foresight into its workings to discover and remove it. The essence of this systematic procedure includes static, dynamic, and hybrid analysis methods. **Static Analysis**, for instance, uses a wide array of tools and techniques to investigate the performance of malware without the original performance. This method offers a comprehensive knowledge of malware and enables scholars to establish effective solutions against it<sup>3</sup>. The enforcement of static analysis is favorable as it excludes the condition of executing possibly malignant files for analysis. Decompilers and disassemblers are some tools that facilitate a direct and accelerated information retrieval technique. However, it is significant that static analysis can obscure virus code, making it laborious to analyze disintegrated code that has been encrypted or disguised. In such cases, **Dynamic Analysis** is an advantageous selection. By accomplishing detrimental processes in a controlled and segregated environment, or sandbox, the dynamic analysis provides a diversity of tools and techniques to grasp the efficacy of malware<sup>2</sup>. It is an effective approach that is applied to achieve greater insight into the authentic performance of an application. This is typically performed by exploring various data sets gathered through malware implementation traces, including system calls, API calls, network activity, and file system activity. One of its core benefits is the potential to escape complications and computing tricks. Regardless of its rewards, dynamic analysis has boundaries, such as low code exposure and prolonged processing delay<sup>4</sup>. As a result, to efficiently investigate malware, a **Hybrid Analysis** approach that weighs static and dynamic analysis is applied. This methodology can overwhelm the disputes presented by dynamic analysis and provide a more extensive consideration of malware activities. The malware examination involves assembling the relevant data to classify and categorize malicious software.

**Signature-based** techniques are normally exercised for their promptness, ease of use, and efficiency in classifying malware. Furthermore, malware analysis techniques are used with signature-based techniques to remove detrimental samples from known malware. This permits straightforward evidence of the existence of these samples in suspicious files. However, signature-based techniques have limitations, such as the difficulty of spotting polymorphic or metamorphic malware and zero-day malware and a high rate of false positives. Moreover, the emergence of metamorphic and polymorphic malware can adapt their code during performance, rendering signature-based methodologies ineffective in recognizing hazardous samples, leading to false negatives amplification<sup>5</sup>.

The field of malware recognition has seen significant developments with the emergence of ML and DL techniques. These techniques have demonstrated success in classifying and categorizing malicious software. ML-based methods are particularly effective in identifying destructive trends in real-world malware occurrences. However, the accuracy of ML models often depends on hand-crafted features, a laborious task that requires dedicated skill. However, DL techniques use procedures that have been precisely proposed for operations such as image processing, speech recognition, and sequence learning. DL models are proficient in spontaneously discovering features from structured and unstructured data, eliminating the need for manual feature extraction. In contrast, DL methods demand massive amounts of data and computing resources to reach high levels of precision<sup>6,7</sup>.

This study establishes a novel multimodal late fusion framework that employs quantitative and imagery data effectively to address the constraints of single-modality and satisfy realistic malware detection specifications. The proposed cutting-edge late fusion methodology boosts adaptability to dynamic malware behaviors while still attaining higher accuracy in identification than the prevailing approaches. The important issues in data integration and model generalizability are addressed by this adaptable and trustworthy approach for evolving threat ecosystems. This study strengthened the field of multimodality by developing the foundation for potential multimodal cybersecurity systems with a concentration on real-world applications in functional domains.

### Novelty of the study

The revolutionary feature of multimodality involves its potential to incorporate multiple formats of data, particularly text, imagery, sound, and sense information, leading to greater awareness of sophisticated activities. It substantially boosts the effectiveness of models by incorporating data from multiple formats, delivering deeper contextual statistics and improved forecasts. However, multimodal models illustrate better adaptability, efficiently addressing distorted or missing data by integrating several techniques. One of the primary advantages is that it encourages cross-modal learning, which contributes to better categorization and transfer learning qualities. The information obtained from one modality may assist in grasping another. Hence, this method facilitates multimodal tasks, which have major implications in practical scenarios such as digital content assessment, self-navigating systems, and clinical inspections.

Furthermore, the multimodal approach encourages better understanding and comprehensibility, especially when integrating modalities like textual or imagery explanations that promote decision-making. To optimize the prospective benefits of each modality, it incorporates attributes, judgments, or embedded information from many sources using advanced data fusion algorithms at three separate levels: early, late, and hybrid. In a nutshell, its potential to deal with sophisticated and complicated concerns renders it particularly appropriate for situations resulting in improved comprehension of events, like digital security solutions motivated by artificial intelligence, establishing it ahead of conventional single-modality systems.

The novelty in the present research is the blend of numerical and visual information, which enhances the decision-making process and the performance of the models. The proposed method delivers greater insight

into the data by integrating numerical data derived from malware patterns or attributes with visual information gathered from malware images. The reliability and precision of the predictive algorithm improved due to this amalgamation, making it practical to discover relationships that would not be evident from a single data type. This distinctive blend of modalities augments potential applications for malware evaluation and identification.

### Challenges

As engineering science evolves, the adaptable malware environment presents a substantial dispute to cybersecurity experts and administrations. Conventional signature-based recognition approaches are progressively portrayed as impractical by the emergent variety and wide array of malware variants. Malware has become more high-tech, exercising methods like heterogeneity to change code and dynamically avoid recognition by static methods. The fight against hazards across numerous platforms is more obscured by environments such as cloud-based services, mobile platforms, and Internet of Things (IoT) devices<sup>8,9</sup>. However, strategies that allow people to securely integrate their Bring Your Own Device (BYOD) personal gadgets associated with company networks are challenging. Regulating malware requires constant awareness to avoid zero-day exploits, focus on the complexity of fileless malware, and identify stealth strategies used to prevent disclosure by traditional antivirus software<sup>10,11</sup>.

Static analytic tools are tricky from the perspective of vague approaches and resource intensity. Dynamic analysis techniques can also encounter limitations from code mystification in addition to the occurrence of false positives and negatives. An effective explanation of a diversified set of complicated data types, such as file content, network activity, and system logs, involves the development of frameworks that have the potential to do so<sup>12</sup>. Additionally, the dynamic performance of malware requires the mining of dynamic features. Furthermore, multimodal data introduces hindrances for simple models regarding generalization, hierarchical representation learning, and dimensionality reduction. These disputes highlight the importance of employing intelligent, flexible and context-sensitive strategies to effectively address the complexity of contemporary virus threats<sup>13</sup>.

### Motivation, aims and impact of study

Cybersecurity breaches have become more prevalent and powerful in the last few decades, with severe implications for real-world incidents. The 2017 WannaCry ransomware strike is one such example<sup>14</sup>. It attacked over 200,000 gadgets in 150 countries, seriously hampering healthcare organizations like the National Health Service (NHS) in the UK and leading to financial obligations of billions of dollars<sup>15</sup>. Similarly, the 2020 SolarWinds incident worldwide affected a range of public and private associations, contributing to serious vulnerabilities and data exposures<sup>16</sup>. The rising rate of these attacks highlighted the growing risk to the banking sector, consumers, and critical systems. These instances emphasize the deficiencies of conventional malware detection techniques, which sometimes overlook sophisticated malware. This research is motivated by the need to create increasingly sophisticated and powerful safeguards against these dynamic threats. This study implies devising a stronger approach for identifying challenging malware by employing multimodal analysis, which incorporates numerical and imagery information. It deals with weaknesses that contribute to real-world problems in important business enterprises<sup>17</sup>.

The following are the aims and objectives of this research:

- To formulate and examine a multimodal malware detection framework that employs late fusion approaches to incorporate numeric and imagery data.
- To strengthen the identification and classification across many different malware types by employing neural networks and ensemble techniques.
- To measure the performance of the proposed multimodal model in addressing intricate malware data when comparing it to unimodal solutions and conventional malware detection techniques.

The findings demonstrate that the proposed multimodal framework performed more efficiently concerning accuracy, precision, recall, and overall robustness.

The multimodal strategy implemented in this research is prospective for better cybersecurity provision and malware detection. The proposed approach optimizes detection capacity, minimizes false positives, and raises the probability of determining advanced malware by incorporating numerical and visual data. The resulting developments could result in safer cyberspaces by offering organizations, the government, and consumers enhanced ways to avoid malicious applications. The rational design of the study has the potential to serve as a standard for multimodal approaches to investigate other cybersecurity domains in the future.

### Background

In the context of DL and artificial intelligence (AI), multimodality illustrates the use of various data formats concurrently, such as textual, pictorial, auditory, sensory and other sources. Multimodal DL frameworks are manufactured to cooperatively examine and illustrate these heterogeneous forms of data to separate and leverage the distinctive information present in each modality. This methodology improves productivity, boosts the comprehension of intricate inputs within the system, and has a broad spectrum of approaches, including sentiment analysis, image captioning, human-computer interaction, medical diagnostics and cyberattack detection<sup>18</sup>.

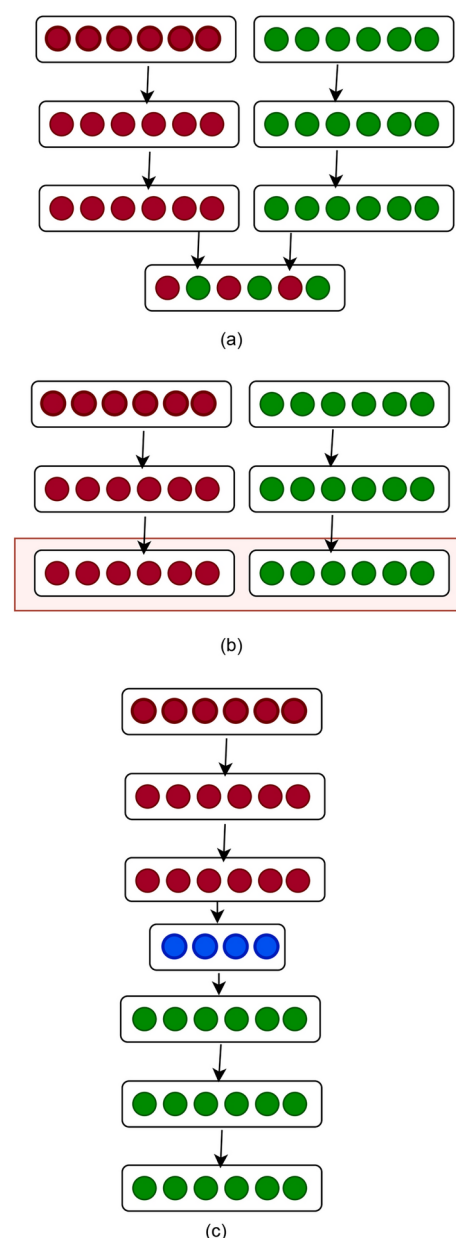
“Multimodality” refers to employing heterogeneous forms of data to gather comprehensive insights. However, this technique is accompanied by particular challenges that investigators and developers should tackle to take advantage of. These challenges include managing data heterogeneity, dealing with modality misalignments, fabricating efficient feature extraction techniques for each modality, amalgamating modalities through suitable fusion strategies, guaranteeing the adaptivity of multimodal systems, dealing with domain adjustment concerns for generalization, and developing appropriate estimation measures that precisely indicate multimodal system

functioning<sup>19</sup>. To address these obstacles, data fusion approaches must be utilized. These methods are categorized into two classes: multimodal fusion representations and multimodal fusion levels<sup>20</sup>.

### Multimodal fusion representations

The idea behind the multimodal system involves employing multiple sensory modalities to process, analyze, and interpret information. The perception of **Marginal Representation** implies that the features and interpretation originated exclusively from a particular modality besides the other modalities. In contrast, **Joint Representation** implicates the incorporation of information from assorted modalities to form an integrated illustration that incorporates the shared characteristics of each modality<sup>21</sup> as illustrated in Fig. 1(a). Furthermore, **Coordinated Representation** involves aligning and designing the representations originating from distinct modalities to quantify each other and offer a regularized perspective of the global contribution. Perceiving these concepts is indispensable to generate effectual and proficient multimodal systems that can manage information from diverse sources<sup>22</sup>.

Coordinated representation uses canonical correlation analysis and cross-modal similarity models to teach discrete but constrained representations for each modality in a coordinated subspace as depicted in Fig. 1(b);



**Fig. 1.** Three categories of deep multimodal representation frameworks. **(a)** Learning a common semantic subspace is the goal of joint representation. **(b)** A coordinated representation framework, subject to certain limitations, learns independent but coordinated representations for every modality. **(c)** An encoder-decoder architecture converts one modality into another while maintaining consistency in their semantics<sup>23</sup>.



Based on the types of constraints, coordinated representation strategies are broadly classified into two classes that are cross-modal correlation-based or cross-modal similarity-based. Cross-modal correlation-based approaches aim to maximize the correlation between representation sets of different modalities, cross-modal similarity-based approaches learn a shared subspace where it is possible to quantify the distance between vectors from different modalities directly<sup>24</sup>.

The last conceptual framework is the encoder-decoder, designed to obtain a transitional representation to change one modality into another. Recently, the encoder-decoder structure has been widely used for multimodal translation tasks, including picture captioning, video description, and image synthesis. These tasks convert between different modalities. As shown in Fig. 1(c), the two main components of an encoder-decoder construction are an encoder and a decoder. The encoder maps a latent vector from the source modality, which the decoder uses to generate a new sample of the target modality. Most configurations of encoder-decoders consist of just one encoder and one decoder; however, certain variants may also contain many encoders or decoders<sup>23</sup>.

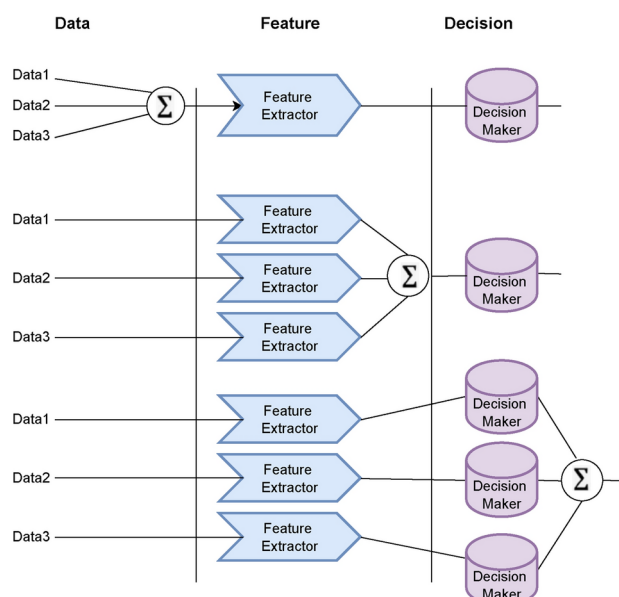
### Multimodal fusion levels

In the domain of multimodal fusion, there exist distinct levels of fusions as demonstrated in Fig. 2, each of which serves an exclusive purpose and presents explicit benefits. These levels of fusion are classified into specific categories as follows: The **Early Fusion/Data-level Fusion** methodology is a comprehensive approach that involves extracting a diverse range of characteristics from multiple modalities<sup>25</sup>. These features can generally be categorized into four classes.

Initially, graphical attributes are subject to color, texture, form, and other factors. These features may include spontaneously established feature points, segmented image blobs, fixed-sized patches or blocks, or the full image. Furthermore, textual traits, consisting of obstructed caption text, video optical character recognition (OCR), production information, and automated speech recognition (ASR) transcripts, are used to extract textual features<sup>26</sup>. Thirdly, audio features, which include the mel-frequency cepstral coefficient (MFCC), fast Fourier transform (FFT), zero crossing rate (ZCR), linear predictive coding (LPC), volume standard deviation, non-silence ratio, spectral centroid, and pitch, are preserved to produce audio features for the fusion method. Finally, motion-related traits, comprising kinetic energy, motion direction and magnitude histograms, optical flows, and motion patterns in specific directions, explain motion-related features in the fusion technique<sup>27,28</sup>.

The simplicity of early fusion is a significant benefit as it does not involve design decisions for mining marginal illustrations. Regardless of its simplicity, early fusion can benefit low-level features to determine cross-modal relationships. However, this methodology may neglect associations that appear only at higher levels of generalization, since marginal representations are not specifically learned. In addition, early fusion approaches may be influenced by the varied sampling rates of each modality, making it difficult to coordinate multimodal features properly. Therefore, it is indispensable that the features to be merged have an identical arrangement before the fusion method. As the number of modalities increases, learning the cross-correlation between different features becomes increasingly challenging<sup>29</sup>.

**Intermediate Fusion/Feature-level Fusion** methodology implicates the convergence of marginal illustrations in feature vectors, rather than using the primary multimodal data. Consistent networks like fully connected or convolutional neural networks are engaged to acquire these representations. In contrast, “heterogeneous” design comprises the aggregation of marginal illustrations through diversified networks. The earlier method is typically used when dealing with similar data types, while the latter approach is better for



**Fig. 2.** Three distinct kinds of fusion data-level, feature-level and decision-level fusion are employed.

diverse multimodal data. There are two types of intermediate fusion: 'marginal' and 'joint.' Marginal fusion involves chaining marginal representations into a classifier, whereas joint fusion involves grasping more abstract joint features. Marginal fusion is also categorized as feature late fusion<sup>26</sup>. Intermediate fusion schemes deliver high adaptability by efficiently merging marginal illustrations in the most favorable arrangement and intensity. This method precisely imitates the true associations between modalities, resulting in the detection of precious joint and marginal latent factors. DL frameworks are predominantly appropriate for intermediate fusion as they allow for the smooth incorporation of marginal representations through a shared layer, while efficiently mapping hierarchical illustrations to real-world scenarios<sup>30,31</sup>.

In the field of ML/DL, the **Late Fusion/Decision-level Fusion** technique refers to the amalgamation of the individual decisions made by unimodal submodels into a final decision. This methodology is favored over merging the primary data or revealed features, as it empowers the discovery of effective marginal illustrations. Additionally, since each model is shaped to fit a specific modality, this approach recommends adaptability to acquire heterogeneous and definite representations.

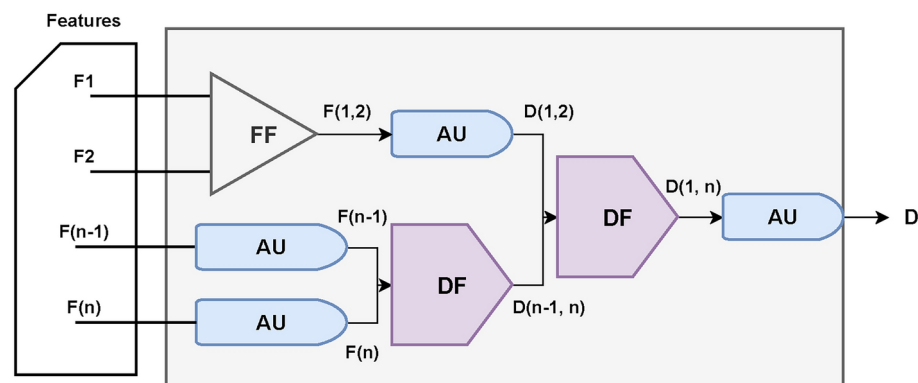
Furthermore, the errors in the submodels are not substantially interrelated, which can lead to complementary results. However, the ultimate model cannot learn multimodal effects at the data or feature level. Late fusion approaches are categorized based on how the sub-model decisions are amalgamated. The forecasts are amalgamated by linking a weighted average or an equal average. Another methodology, "meta-learning," involves an ML model to make absolute forecasts using the prediction probabilities as input<sup>32</sup>.

Generally, the late fusion method has been demonstrated to be a useful methodology in machine learning for various applications. By adapting each model to fit an explicit modality, this methodology can achieve more precise forecasting while diminishing errors. Furthermore, the ability to merge complementary results of the submodels results in an overall improvement in the precision of the final model. The latter has numerous benefits when evaluating the feature-level fusion technique to decision-level fusion. Although feature-level fusion utilizes features from different modalities that often have diverse illustrations, decisions made at the semantic level have identical significance. This makes decision merging a simpler task. Additionally, decision-level fusion offers scalability in the modalities used in the fusion process, which is more difficult to achieve with feature-level fusion. Another benefit of the late fusion methodology is that it employs the best techniques to evaluate each modality, such as the support vector machine (SVM) for images and hidden Markov models (HMM) for audio<sup>33</sup>. This leads to considerably greater adaptability than early fusion techniques. On the other hand, the weakness of the late fusion method is that it does not derive the benefit of the feature-level correspondence between modalities. Furthermore, discovering new classifiers remains time-consuming and laborious because it demands multiple classifiers to acquire local decisions<sup>21</sup>.

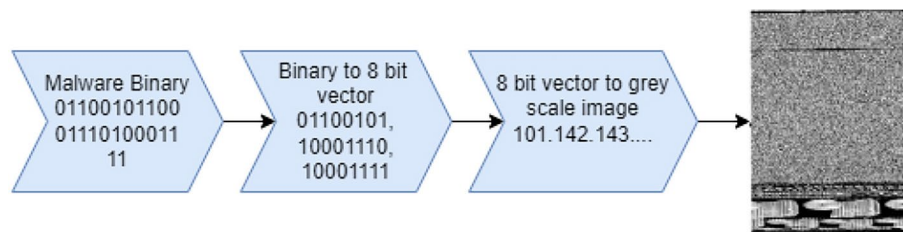
**Hybrid Multimodal Fusion** is an effective approach that incorporates feature-level and decision-level fusion methods into a single hybrid approach. This technique improves the benefits of both methods. In this technique, an analysis unit (AU) scrutinizes the feature vector after a feature fusion (FF) unit has integrated the features. Moreover, individual AUs inspect specific features, while a decision fusion (DF) unit combines their decisions. Eventually, a final decision is extended by fusing the conclusions from the earlier phase through a DF as depicted in Fig. 3. Employing a hybrid fusion strategy enables combining the advantages of early and late fusion procedures<sup>30</sup>.

### Malware visualization

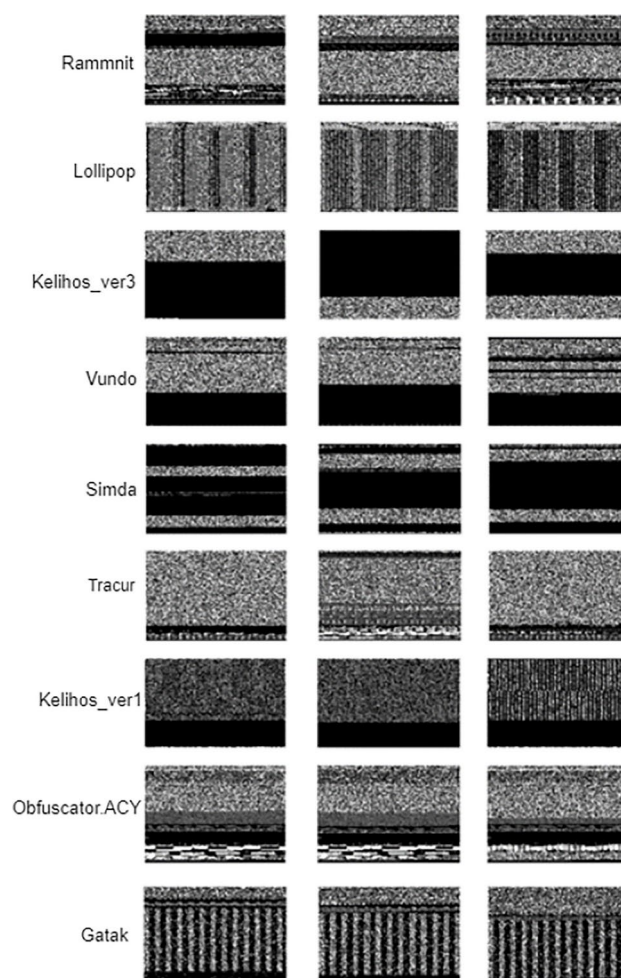
Visual malware illustration is an effective methodology to acquire a deeper understanding of its patterns and behavior. This approach proposes substantial understanding by modifying sophisticated, high-dimensional data into an understandable visual presentation. Malware tends to be represented in imagery and numerical formats in this context, offering exceptional benefits for analysis and classification. Different malware essentials, such as file properties, opcode sequences, and network traffic statistics, are classically expressed as numerical datasets, allowing for comprehensive statistical analysis and feature extraction. Furthermore, imaging databases translate malware binaries into grayscale or byte plot images, explaining the structural patterns of malware code. Each byte demonstrates a distinct pixel allocated to identify an image indicating a virus. The causative array must



**Fig. 3.** Hybrid multimodal fusion implementation.



**Fig. 4.** Visualizing malware as an image<sup>35</sup>.

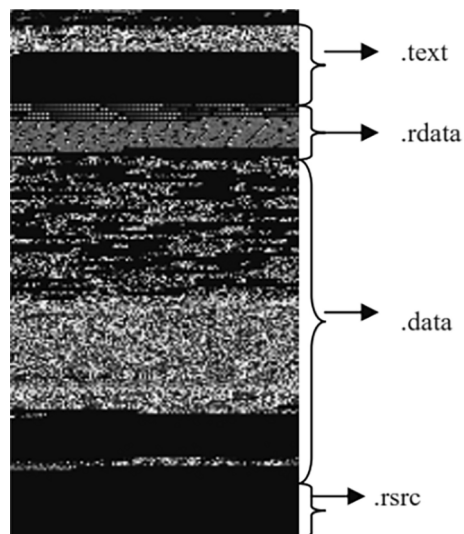


**Fig. 5.** Gray-scale pictures of nine malware families<sup>36</sup>.

then be structured as a 2-D array and visualized as a grayscale image. The values fall on the [0,255] scale from 0 (black) to 255 (white). Figure 5 displays gray-scale illustrations corresponding to nine malware families.

The visual representations of software executables associated with diverse groups fluctuate visually despite their visual correspondence. Demonstrating a malicious executable as an image makes it simpler to distinguish between binary components<sup>34</sup>. Furthermore, malware authors are inclined to barely adapt a minor fraction of the code to construct novel variants. Consequently, new binaries created by reprocessing early malware would be analogous. Figure 4 provides a precise scheme for generating visual representation from binary malware elements.

Moreover, it is probable to recognize small amendments in malware while preserving the general configuration of instances from the same group when it is characterized as an image. When carefully analyzed, one can typically categorize various software sections, with distinctive feature arrangements. Figure 6 illustrates a sample image of a prevalent Trojan downloader named “Dontovo.A.” This malicious computer software downloads and executes arbitrary files.



**Fig. 6.** Various sections of Trojan: Dontovo.A<sup>37</sup>.

Interestingly, different segments or binary malware segments demonstrate distinctive image textures. The .text section holds the executable code. It is evident from Fig. 6 that the preliminary part of the .text section comprises finely-grained textured code. The outstanding portion of this segment is filled with zeros (black), indicating zero padding. The subsequent .data section encompasses uninitialized code (black patch) and initialized data (finely-grained texture). Finally, the .rsrc fragment grasps all the module resources, incorporating icons that applications may use. This dual approach leverages the strengths of traditional numerical analysis and the powerful pattern recognition capabilities of deep neural networks (DNN), requiring a comprehensive framework for robust malware detection and classification.

### Literature review

Independent utilization of textual, quantitative, and imagery features is the primary goal of the prevalent literature on ML and DL methodologies in cybersecurity, specifically for virus detection. An array of studies has examined the descriptive characteristics, including log files, application programming interface (API) calls, and network-related data, as Table 1 illustrates. A considerable amount of exploration has been carried out regarding using quantitative characteristics, such as feature vectors from static analysis, behavioral metrics, and statistical analysis of network traffic, to identify security hazards. The Utilization of imagery features is gaining popularity as an individual way of inspecting malware arrangements. A typical example is image-based malware detection, which employs binary files turned into grayscale images. These multiple methods each contribute distinct perspectives on several aspects of cybersecurity concerns.

Only a few investigations have thoroughly implemented multimodal strategies amalgamating textual, quantitative, and imagery data. Multimodal methodologies are investigated to some extent, considering that such investigations show benefits such as optimized efficiency of models, adaptability, and the power to grasp complex patterns across many data types<sup>38,39</sup>. There is an evident lack of investigation in this domain that methodically integrates various forms of data to develop perfect cybersecurity solutions. Most of the exploration carried out in this field remains focused on distinctive modalities. This study emphasizes the requirement to perform a more thorough investigation into how integrating different modalities could strengthen security measures. There is also a great potential for new advancements in multimodal deep learning for future research.

The existing studies have underlined limitations in their corresponding malware identification methodologies. In<sup>40</sup>, the authors employed a scheme that applied data from two scenarios network-telemetry and endpoint-telemetry but were unable to fuse additional data types. In<sup>41</sup>, the recommended work was limited in representing information to identify Android malware. Likewise<sup>42</sup>, attempted multimodality by obtaining features from unzipped APK files (manifest files, dex files, and shared libraries) but was limited in the variety of data integration. The scholars of<sup>43</sup> reported that the dynamic investigation of malware based on software features seeks to evade techniques, and there is room for advancement in powerful hardware-based recognition, specifically in classification accuracy. <sup>18</sup> depended on a selected array of static features, which are susceptible to obfuscation techniques used by malware.

Moreover<sup>44</sup>, amalgamated arithmetic, contextual, and graph mining features but was limited in the integration of other significant feature types. Further<sup>45</sup>, pointed out the lack of their DroidFusion framework in expanding the algorithm to deal with multiclass classification and<sup>2</sup> employed only two modalities for file data (JSON and DEX), limiting the possibility of other potential data types. In<sup>46</sup>, manually extracted features were consumed, while the scholars highlighted the limitation that machine-generated features could cause better performance. Studies<sup>47</sup> and<sup>49</sup> utilized only unified data, limiting the diversity of characteristics. <sup>48</sup> integrated structural, semantic, and permission-based features but lacked the integration of other important features that could improve accuracy. The scholars of<sup>49</sup> debated the need to advance the extensibility of GSEDroid, optimize

Ref/Year	Framework	Modalities	Dataset	Malware attack		Research gap		Accuracy
				Android	Windows	Textual fusion	Visual fusion	
40	Theoretical multimodal framework	–	–	☐	✓	☐	☐	–
41	Principal Component Analysis (PCA) and Co-relation attribute evaluation (CAE)	Android APK file (Unimodal)	Android Proguard, Drebin, Open-source apps, Kharon, Androzoo, CICAAGM	☐	✓	☐	☐	95%
42	Deep Neural Network (DNN)	Features extracted from the APK file (Unimodal)	Omnidroid dataset	✓	☐	☐	☐	good
43	DNN	API call sequences and the grayscale images mapped by the memory dump square (multimodal)	OpenMalware VirusTotal	☐	✓	✓	✓	93.5%
18	Late integration multi-modal approaches	Four types of features are Byte Entropies, Address Table, String Hashes, and Portable Executable (PE.) headers (Multimodal)	Berlin and Saxe	☐	✓	✓	☐	good
44	Different clustering sequences	API calls, statistical, contextual, and graph mining features (Unimodal)	malsource	✓	✓	☐	☐	99.7%
45	Multilevel classifier fusion approach (DroidFusion)	Unimodal features	Drebin- 215 and Malgenome- 215, McAfee- 350 and McAfee- 100	✓	☐	☐	☐	good
2	ML and DL frameworks–intermediate fusion	Features from JSON and DEX files (Multimodal)	Omnidroid	✓	☐	✓	☐	good
46	ML and DL frameworks–feature fusion	handcrafted features (Multimodal)	DREBIN	✓	☐	✓	☐	93.24%
47	Clustering, classification and blockchain	Unimodal data	Google Play Store and Chinese App Store	✓	☐	☐	☐	91.88%, 99.36%
48	Deep multimodal feature fusion	Structural and semantic features (Multimodal)	Three constructed datasets	✓	☐	✓	☐	97.28%, 99.54%
49	GSEDroid: lightweight embedding model based on CodeBERT and TextCNN	API call graph with permission and opcode semantic features (Unimodal)	–	✓	☐	☐	☐	99.47%
50	Multimodal deep learning model (MDLDroid)	System calls and dynamic API features (Multimodal)	Own developed dataset	✓	☐	✓	☐	good
51	CNN	Malware images and structural entropies (Unimodal)	Kaggle Microsoft Malware Classification, Maling, and BODMAS datasets	☐	✓	☐	✓	good
52	CNN-BiLSTM	Char2 Vec, Token2 Vec, AST2 Vec, and Rela2 Vec (Unimodal)	Own dataset	☐	✓	✓	☐	94.02%
53	CNN, LSTM, RNN, GRU	Packet Capture (PCAP) files (Multimodal)	CIC-IoT 2022, CIC-IoT 2023, and Edge-IIoT	✓	☐	✓	✓	98.2%
54	CNN, VGG16	Three different visual malware features: Grayscale Image, Entropy Graph, and SimHash Image (Multimodal)	BIG2015	✓	☐	☐	✓	98.2%
55	Support Vector Machine (SVM) and XGBoost	(Unimodal)	Own dataset	☐	✓	✓	☐	99.6%
56	RNN, LSTM	Twitter data (Unimodal)	Own dataset	☐	✓	✓	☐	94%

**Table 1.** Comprehensive literature review to identify potential gaps in existing research.

the model and graph size, and combine adversarial graph-based techniques. Lastly<sup>51,52,55,56</sup>, were limited to unimodal data, while<sup>50,53,54</sup> utilized multimodality but only within the same data type, without the opportunity to improve capacity through diverse data integration.

In<sup>53</sup>, the scholar delivers a 99% accuracy for a sophisticated intrusion detection system. It provides safety by utilizing multimodal big data and transfer learning to improve assault detection. The approach leverages Spark to streamline large-scale data, word2vec to obtain semantic features, and ResNet to examine the texture of network data stored as images. In<sup>57</sup>, instigators do not allow the fusion of image features and apply feature fusion and late fusion to various textual features, including strings, integers (five binary features), floating points, and timestamps. In<sup>58</sup>, authors also employ MobileNetv2 to retrieve texture and structural data, transform binary files into 2D pictures, and apply voting-based algorithms for late fusion to offer malware detection. It does not include various data modalities.

This research advances previous techniques by employing multimodal fusion, which incorporates both quantitative and imagery data to boost cyberattack identification, verify findings and maximize the influence of the proposed approach. The proposed strategy amalgamates these numeric and visual modalities to generate a comprehensive set of characteristics that significantly enhances the effectiveness of detection compared to prior investigations in which the fusion focused merely on multiple quantitative data types<sup>57</sup> or texture features extracted from malware images<sup>58</sup>. By incorporating structural and behavioral nuances across several data types, the privilege of the multimodal fusion approach of this study outperforms conventional methods. It



offers improved and accurate threat diagnosis in sophisticated system contexts. This progress is significantly complemented by including recent research in the literature review, which will demonstrate the exclusive ability of this technique to facilitate multimodal malware data and effective learning capacity.

Research gap

The extensive review of the literature highlights that existing studies usually focus on static or dynamic characteristics. Although recent developments in cybersecurity involve ML and DL frameworks, few leverage a comprehensive late fusion strategy to improve malware classification accuracy. There is a lack of an efficient multimodal framework that integrates both numeric and imagery data representations for the Android and Windows platforms. Most of the prevailing studies focus on various textual or visual features individually, limiting their adaptability against evasive malware techniques. This study bridges the gap by introducing a state-of-the-art multimodal framework that integrates quantitative and imagery features to perform outstandingly for dynamically evolving malware.

Datasets

The subsequent section is divided into two subsections: one determines the numerical data, and the other is the imaging data. Together, they provide a comprehensive summary of both datasets used in this study. The first data source is the CCCS-CIC-AndMal-2020 dataset, which includes comprehensive numerical information extracted from malware behavior and network traffic. Moreover, the blended image dataset encompasses visual attributes derived from binary malware visualizations. These two datasets present massive opportunities to investigate the ML and DL frameworks for effective malware detection. By integrating numerical and visual inputs, this research aims to strengthen the precision and robustness of detection models.

Blended malware image dataset

The technique of categorizing malware encompasses transforming PE files into byteplot images. Malevis and Maling are two datasets used for this image categorization. Table 2 and Table 3 illustrates the class distribution across both data sets. The class distribution in the Maling dataset is considerably out of balance, contrary to the well-balanced Malevis dataset. To address this issue, a single dataset is created by amalgamating the five classes from the Maling dataset and all categories from the Malevis. The key objective of this data set is to accommodate both RGB and grayscale byteplot images in the same dataset and to facilitate the multiclass classification of malware byteplot images<sup>59,60</sup>. The Malevis dataset aimed to provide scholars with an RGB-based ground truth

ClassID	Family	Malware category	Sample no.
1	Adposhel	Adware	350
2	Agent	Trojan	350
3	Allaple	Worm	350
4	Amonetize	Adware	350
5	Androm	Backdoor	350
6	Autorun	Worm	350
7	BrowseFox	Adware	350
8	Dinwod	Trojan	350
9	Elex	Trojan	350
10	Expiro	Virus	350
11	Fasong	Trojan	350
12	HackKMS	Riskware	350
13	Hlux	Worm	350
14	Injector	Trojan	350
15	InstallCore	Adware	350
16	Multiplug	Adware	350
17	Neoreklami	Adware	350
18	Neshta	Virus	350
19	Other	–	350
20	Regrun	Trojan	350
21	Sality	Virus	350
22	Snarasite	Trojan	350
23	Stantinko	Trojan	350
24	VBA	Macro Malwares	350
25	VBKrypt	Trojan	350
26	Vilsel	Trojan	350
	Total	–	9100

Table 2. Category wise malware instances for Malevis dataset<sup>34</sup>.

ClassID	Family	Malware category	Sample no.
1	Adialer.C	Dialer	122
2	Agent.FYI	Backdoor	116
3	Allaple.A	Worm	2949
4	Allaple.L	Worm	1591
5	Alueron.gen!J	Trojan	198
6	Autorun.K	Worm	106
7	C2LOP.P	Trojan	200
8	C2LOP.gen!g	Trojan	146
9	Dialplatform.B	Dialer	177
10	Dontovo.A	Downloader	162
11	Fakerean	rogue	381
12	Instantaccess	Dialer	431
13	Lolyda.AA1	PWS	213
14	Lolyda.AA2	PWS	184
15	Lolyda.AA3	PWS	123
16	Lolyda.AT	PWS	159
17	Malex.gen!J	Trojan	136
18	Obfuscator.AD	Downloader	142
19	Rbot!gen	Backdoor	158
20	Skintrim.N	Trojan	80
21	Swizzor.gen!E	Downloader	128
22	Swizzor.gen!I	Downloader	132
23	VB.AT	Worm	408
24	Wintrim.BX	Downloader	97
25	Yuner.A	Worm	800
	Total	–	9339

**Table 3.** Category wise malware instances for Mallmg dataset<sup>35,59</sup>.

dataset for evaluating their image-based multi-label malware recognition exploration. An array of 26 (25 + 1) class-byte images is stipulated for this rationale. In such a situation, one class stands for the “legitimate” samples, whereas the additional twenty-five classes illustrate numerous malware subtypes. Initially, Sultanik’s bin2png software separates the binary pictures in a 3-channel RGB arrangement from malware records to create this corpus. The photographs were vertically long and ranged from two definite square resolutions (224x224 and 300x300 pixels). The Malevis dataset comprises 9100 training and 5126 validation RGB images.

All training classes encompass 350 image samples, while the validation set has a heterogeneous number of images. Since the temperament of malware detection is formulated to discriminate legitimate ones from malware, a relatively significant number of “legitimate” instances are offered for the validation example (350 vs 1482)<sup>34,62</sup>.

### CCCS-CIC-AndMal- 2020

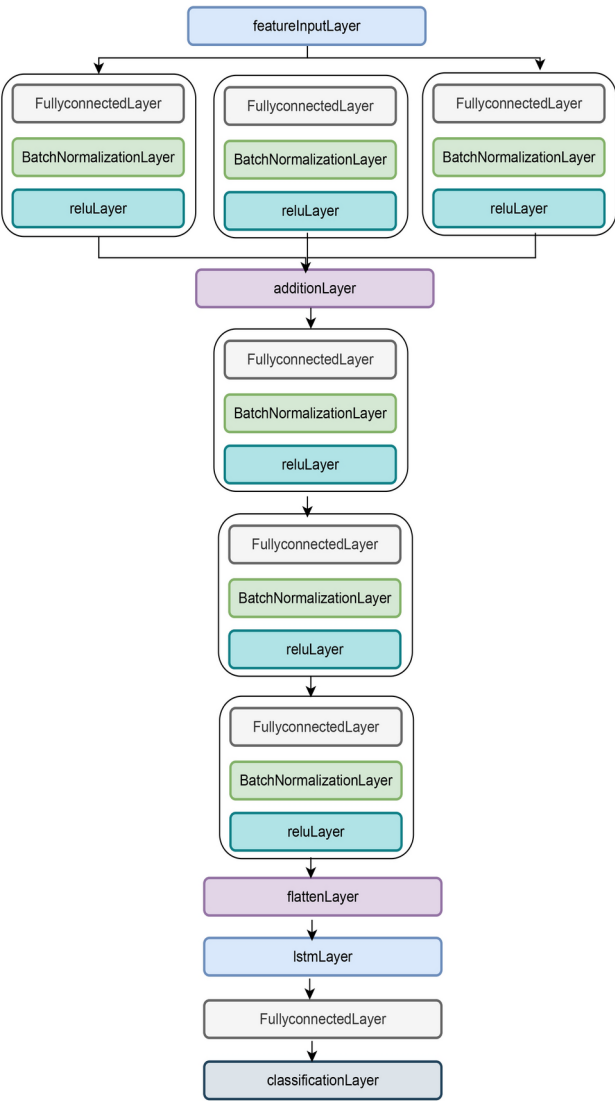
A large Android malware dataset known as CCCS-CIC-AndMal- 2020 comprises 400 K Android app samples, 200 K of which are malicious and 200 K of which are benign. There are 14 popular malware categories and 191 prominent malware families. The Canadian Institute for Cybersecurity (CIC), in collaboration with the Canadian Centre for Cyber Security (CCCS), gathered two hundred thousand Android malware apps, labeled and classified according to their families, to create a typical dataset. The 200 K benign Android apps of the Androzoo dataset are gathered to balance the massive dataset. They collected fourteen different forms of malware. CCCS collaborated with CIC to collect and analyze real-world Android malware apps<sup>63</sup>. The VirusTotal dataset is utilized to classify malware families based on the agreement of 70% anti-viruses. Table 4 exhibits details of 14 different classes of Android malware, including the number of instances and types within each category.

### Proposed model

The proposed topology incorporates two evident DL models, CNN and LSTM, in different arrangements to fabricate a 24-layer framework. Figure 7 portrays the detailed organization and arrangement of the layers. The layers are organized in parallel and series fashion to optimize model performance. The most effective layer configuration is established through experimental progressions instead of precise observance of established norms. The model embarks with a feature input layer, based on which the input is granted as numeric features. Afterward, the input travels three sets of parallel-arranged convolution bases, each comprising a fully connected layer, batch normalization and ReLU layer. These convolution bases perform low-level feature extraction. The feature matrices extracted from the initial convolution bases are aggregated using an additional layer and then passed through three additional convolution bases arranged sequentially. Once the convolutional bases have identified sufficient feature details, the resulting feature map is passed to a flatten layer, which is then

ClassID	Category	No. of families	No. of samples
1	Adware	48	47,210
2	Backdoor	11	1,538
3	File Infector	5	669
4	No Category	–	2,296
5	PUA	8	2,051
6	Ransomware	8	6,202
7	Riskware	21	97,349
8	Scareware	3	1,556
9	Trojan	45	13,559
10	Trojan-Banker	11	887
11	Trojan-Dropper	9	2,302
12	Trojan-SMS	11	3,125
13	Trojan-Spy	11	3,540
14	Zero-day	–	13,340

**Table 4.** Category wise malware instances for CCCS-CIC-AndMal- 2020 dataset<sup>61</sup>.



**Fig. 7.** Proposed ensemble deep learning framework.

connected to the LSTM layers. Eventually, the output of the LSTM layers is directed to a classifier responsible for identifying malware. The architectural framework used to analyze malware imagery datasets is based on a consistent structure but with specific accommodations addressing image data attributes. In this framework, convolution bases are deployed using convolution2D layers instead of fully connected layers, and an image input layer is exclusively designated for the input layer. Furthermore, following each ReLU layer within the convolution base, adding a max-pooling layer is indispensable to enhance the feature maps further. These developments result in extracting relevant features that are subsequently added using a concatenation layer to capture the comprehensive representation of the input data.

Figure 8 depicts the block diagram of the proposed late-fusion model. It illustrates the multimodal analysis-based malware detection technique incorporating quantitative (CSV) and visual data. The subsequent workflow is demonstrated, commencing with data preprocessing and feature extraction. Then, it advances with classification via RUSBoost and an ensemble deep neural network. Both models are trained independently on individual datasets. Following the training phase, this system takes advantage of the late fusion approach, which involves the results obtained from the proposed neural network model. RUSBoost is incorporated through ensemble techniques, such as majority vote, stacked, boosted and bagged ensembles. By aggregating the best features of each data modality, this amalgamation enhances decision-making and generates more reliable malware detection observations. The integrated output is a foundation for the final decision, guaranteeing extensive investigation by incorporating quantitative and visual data.

### Methametrical explanation of proposed model

The technical explanation of the proposed model is established through the detailed mathematical formulation against transformations occurring at each step. For the contributed input feature matrix  $X$ , the convolutional layers employ a set of kernels  $W$  with biases  $b$ , operating the process which is depicted in (1):

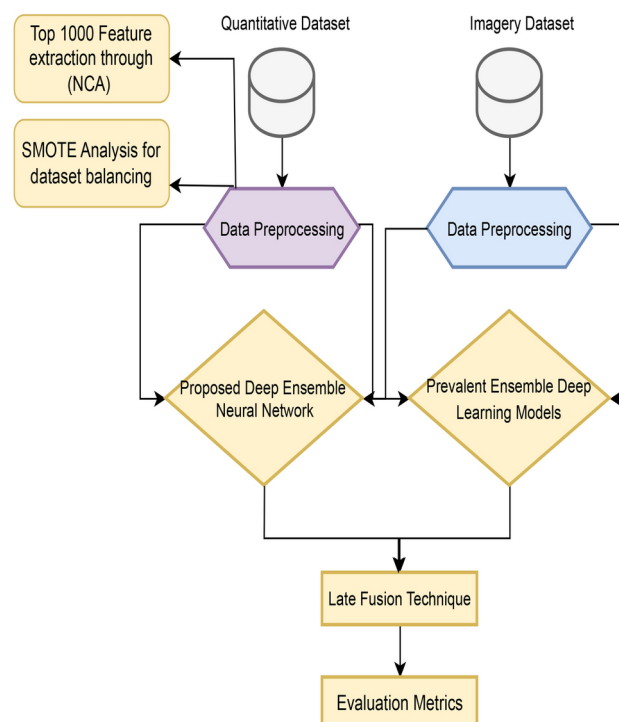
$$F_i = \sigma(W_i * X + b_i) \quad (1)$$

where  $F_i$  signifies the feature maps generated by the  $i$ -th convolutional layer,  $*$  indicates the convolution operation, and  $\sigma$  is the activation function, typically ReLU as represented in (2):

$$\sigma(x) = \max(0, x) \quad (2)$$

The extracted feature maps from the primary convolution bases are amalgamated using an addition function<sup>64</sup> as illustrated in (3):

$$F_{\text{combined}} = \sum_{i=1}^n F_i \quad (3)$$



**Fig. 8.** Block diagram of proposed multimodal late fusion framework.

where  $n$  is the number of parallel convolution bases. The collective features are then handled by supplementary convolutional layers, observed by a flattening operation that restructures the output into a vector  $v$ .

$$h_t = \phi(W_h h_{t-1} + W_x v_t + b_h) \tag{4}$$

The LSTM layers practice the sequential dependencies of the extracted features through the subsequent adaptations<sup>65</sup> as shown in (4) :

where  $h_t$  is the hidden state at time stamp  $t$ ,  $W_h$  and  $W_x$  are weight matrices,  $b_h$  is the bias, and  $\phi$  is a non-linear activation function such as the sigmoid or tanh function. In conclusion, the (5) represented that the classifier layer calculates the probability of each class applying the softmax function:

$$P(y = j|X) = \frac{e^{W_j h_T}}{\sum_{k=1}^K e^{W_k h_T}} \tag{5}$$

where  $P(y = j|X)$  signifies the probability of class  $j$ ,  $W_j$  are the weights for the  $j$ -th class,  $h_T$  is the final LSTM hidden state, and  $K$  is the number of malware categories. This mathematical presentation presents a methodical understanding of the feature extraction, conversion, and classification processes in the proposed model.

Evaluation metrics

The evaluation methods for computing the implementation analysis of the proposed framework are as follows:

- Accuracy: It quantifies the percentage of accurately classified occurrences among the total illustrations. It is commonly used and may present confusing scores for imbalanced datasets. (True Positives + True Negatives)/Total Instances
- Precision: It determines the rate of true positive forecasts among all positive predictions. It is beneficial for estimating the significance of the positive predictions made by the framework. True Positives/(True Positives+False Positives)
- Recall (Sensitivity): It computes the percentage of true positive predictions from all genuine positive illustrations. It effectively evaluates the ability of the model to classify all significant instances. True Positives/(True Positives+False Negatives)
- F1-Score: It is the harmonic mean of precision and recall. It delivers a single metric that stabilizes precision and recall, which benefits imbalanced datasets.  $2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$
- Confusion Matrix: A confusion matrix is a table that is exploited to depict the performance of a classification model by evaluating the genuine and anticipated classifications. It stipulates a comprehensive understanding of the various errors generated by the model. Components: True Positives, True Negatives, False Positives, False Negatives.

Experiment and analysis

This section employed the experiments fabricated to gauge the performance of the proposed ensemble deep model and the implementation of the late fusion technique against it. The system is implemented in Matlab 2022, exploiting the Deep Network Designer feature to develop the framework. The performance of this framework is evaluated on an exhaustive experimental and comparative analysis of several different ensemble algorithms, such as RUSBoost, AdaBoostM2, Subspace, Random Forest, and BagTree. Initially, the efficiency of these models is assessed individually by employing them independently to visual and quantitative datasets. Subsequently, the findings generated from all modalities have been integrated across all ensemble models using late fusion approaches. An array of methods, including majority voting, stacking, bagging, and boosting, were implemented to measure the performance of this fusion. Moreover, the RUSBoost classifier to address the class imbalance and the proposed neural network were subjected to late fusion after being utilized individually for each dataset. The efficiency is augmented by employing the complementary benefits of the imagery and quantitative data. The experiments are performed with the parametric values discussed Table 5: The experiments were performed on powerful computing resources to process the multimodal malware dataset. This system has a 7th-generation Core i5 processor with 16 GB of RAM to ensure efficient computation and model execution. The effectiveness of the model is computed against the key evaluation metrics, including accuracy, precision, recall, and F1-score. The key to constructing a powerful classification framework is eradicating unnecessary, inappropriate, or noisy

Ensemble models	For numeric dataset			For imagery dataset		
	Epoch	Batch size	Learning rate	Epoch	Batch size	Learning rate
Proposed Ensemble	100	150	0.001	500	150	0.003
Rusboost	3000	-	0.1	4000	-	0.004
Random Forest	300	-	-	3000	-	-
Subspace	3000	-	-	3000	-	-
AdaboostM2	3000	-	0.1	3000	-	0.1
BagTree	3000	-	-	3000	-	-

Table 5. Hyperparameters for proposed ensemble deep NNW and ensemble models.



features and preserving only the top 1000 discriminatory features for the quantitative dataset. The experimental procedure is divided into three phases.

### First phase: data preprocessing

The primary phase of the experiment includes data preprocessing on imagery and quantitative datasets. This phase follows the following steps to accomplish data preprocessing:

**Step 1: Feature Extraction (Imagery)** The feature extraction phase of the imagery data involves the conversion of raw image data into significant numerical features through convolutional operations or numerical conversions<sup>64</sup>, as discussed in (6).

$$F(i, j) = \sum_{m=-k}^k \sum_{n=-k}^k I(i-m, j-n) \cdot K(m, n) \quad (6)$$

where:

- $I(i, j)$  denotes the grayscale image pixel intensity at location  $(i, j)$ .
- $K(m, n)$  is the convolution kernel.
- $F(i, j)$  is the extracted feature at that position.

**Step 2: Feature Selection (Quantitative)** The feature selection for the CCCS-CIC-AndMal- 2020 dataset through NCA. The tabular CIC dataset comprises a huge number of features. NCA is applied to extract and pick numerical features, concentrating on the top 1000 features to reduce dimensionality. NCA is a supervised feature selection technique. It plays a key role in distinguishing the most significant features by ranking the most appropriate characteristics. It learns a feature-weighting matrix by optimizing the leave-one-out classification performance, thus advancing the classifier's ability to discriminate between distinct classes. Given a dataset with  $N$  samples and feature vectors  $x_i \in \mathbb{R}^d$ , NCA allocates a probability that a contributed sample  $x_i$  picks another sample  $x_j$  (of the same class) as its neighbor<sup>66</sup> and elaborated in (7):

$$p_{ij} = \frac{\exp(-\|Wx_i - Wx_j\|^2)}{\sum_{k \neq i} \exp(-\|Wx_i - Wx_k\|^2)} \quad (7)$$

where:

- $W$  is a learned transformation matrix that allocates rank to each feature.
- $\|Wx_i - Wx_j\|^2$  calculates the squared Euclidean distance between feature vectors after restructuring.
- The idea is to augment the probability of valid categorization by adjusting  $W$ , efficiently choosing the most significant features.

Features with weights close to zero are rejected.

**Step 3: SMOTE Analysis)** The SMOTE technique is used to produce artificial illustrations for marginal categories due to the highly imbalanced nature of the quantitative data set. This will guarantee that the proposed framework has adequate data to learn from the minority classes. To increase the overall precision and accuracy of the framework, it is indispensable that it can precisely classify the underrepresented classes. For a minority class sample  $x_i$ , a new synthetic sample  $x_{\text{new}}$  is generated and demonstrated by (8):

$$x_{\text{new}} = x_i + \lambda \times (x_j - x_i) \quad (8)$$

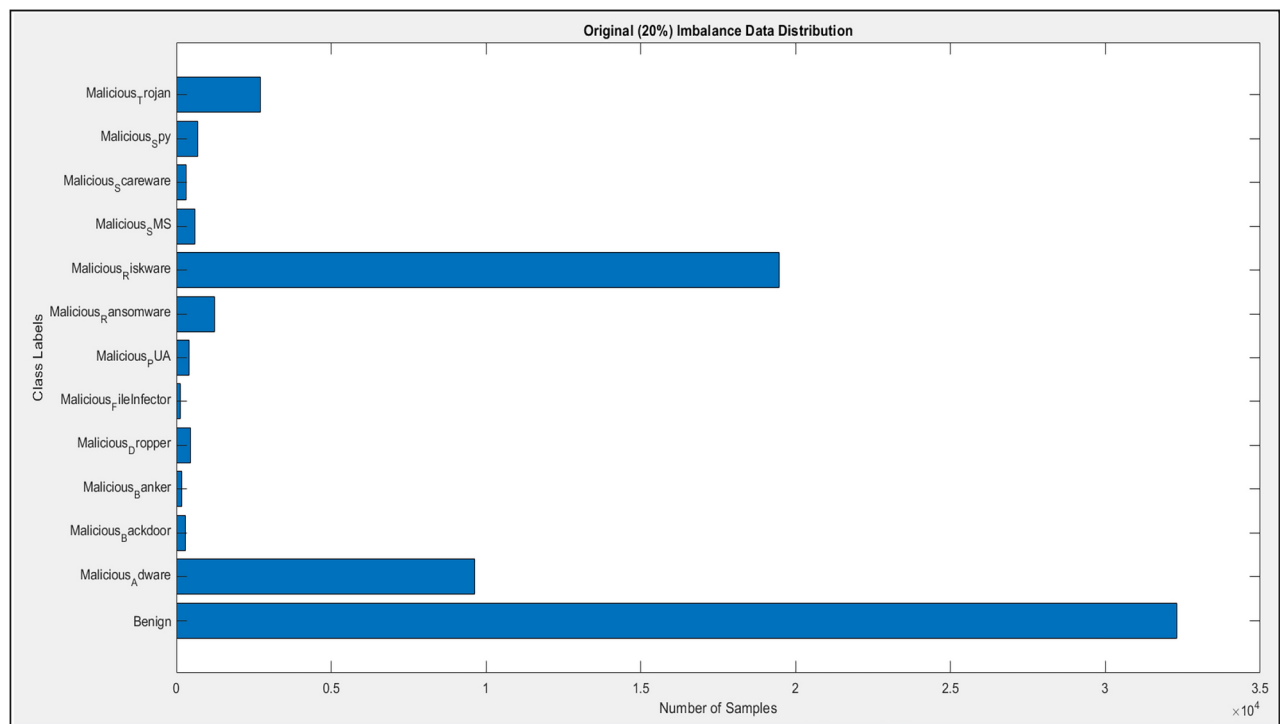
where:

- $x_i$  and  $x_j$  are two actual feature vectors of the minority class.
- $\lambda$  is a random number in the range  $[0, 1]$ .
- The new sample  $x_{\text{new}}$  lies along the line segment between  $x_i$  and  $x_j$ , guaranteeing variety in artificial samples.

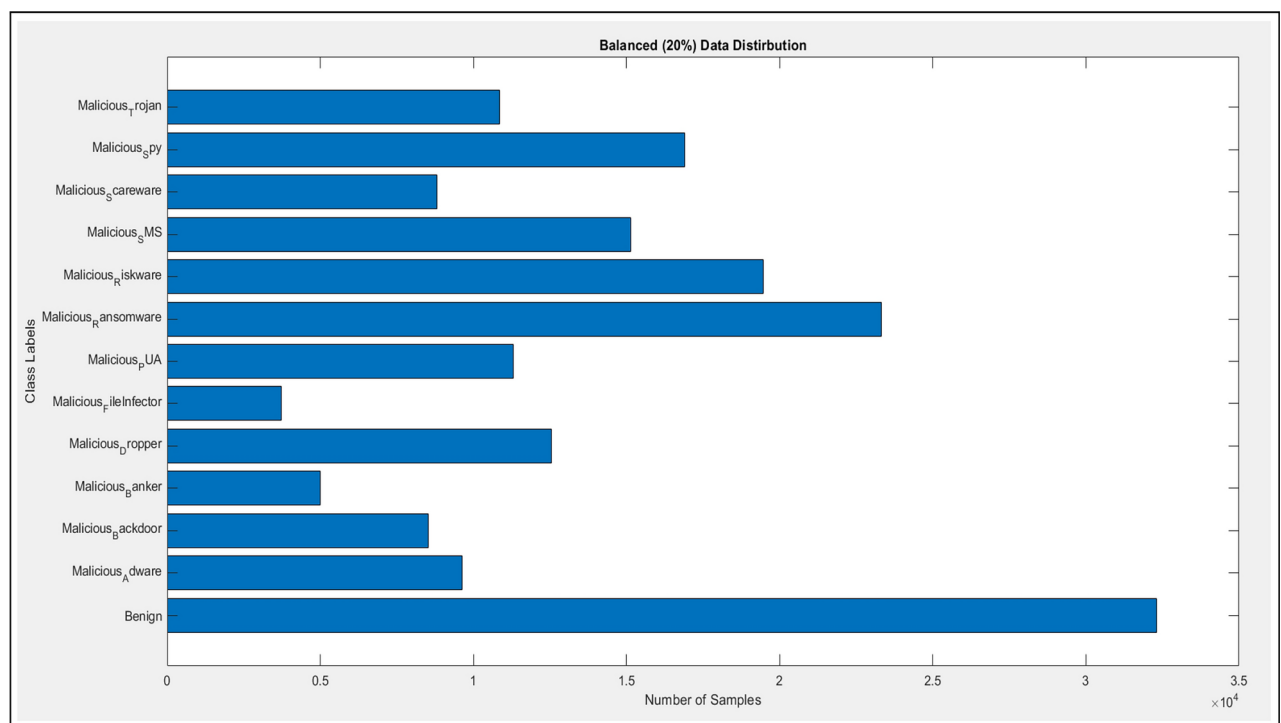
As the dataset is huge and multidimensional, only 20% (68433 samples) of the data with the top 1000 features is designated for processing to accommodate the resource constraint environment. Hence, data balancing is performed on the selected data. Figure 9 demonstrates the division of the number of samples between each class in the selected dataset, while Fig. 10 exhibits an illustrative demonstration of the data set after using SMOTE (177541 samples), highlighting the class balancing effect.

### Second phase: training for unimodal datasets

The second phase involves the application of the proposed deep ensemble neural network for the numeric CCCS-CIC-AndMal- 2020 and the blended malware imagery datasets. Initially, SMOTE analysis is performed on the numeric dataset due to its imbalanced nature. The proposed framework is employed on the SMOTED CCCS-CIC-AndMal- 2020 dataset to exploit the powers of synthetic oversampling to manage class imbalance. The SMOTE technique handles minority classes by generating artificial samples, thus stabilizing the dataset and advancing the framework to learn from underrepresented classes. Afterward, the SMOTED dataset is split into training sets (70%) and testing sets (30%) and evaluated against the deep ensemble network. The proposed



**Fig. 9.** Original data distribution for CCCS-CIC-AndMal- 2020 dataset.



**Fig. 10.** Balanced data distribution for CCCS-CIC-AndMal- 2020 dataset after SMOTE analysis.

network is trained using the 'Adam' optimizer and specific hyperparameters, including learning rate (0.001), 100 epochs and mini-batch size (150).

A training graph and confusion matrix, depicted in Figs. 11 and 12, respectively, measure the effectiveness of the model. The comparative analysis of the proposed neural network with the prevailing observations for the CCCS-CIC-AndMal- 2020 dataset across separate malware classes is illustrated in Table 6. The accuracy of the

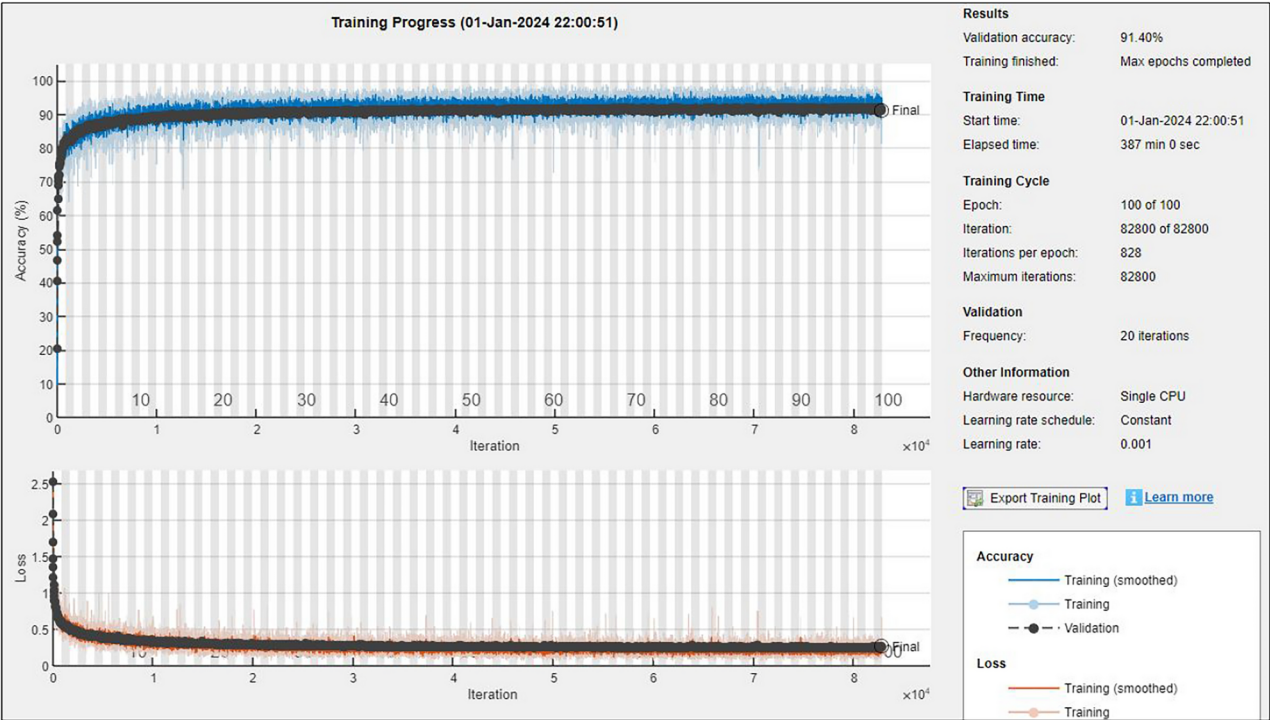


Fig. 11. Training result of CCCS-CIC-AndMal- 2020 dataset using proposed deep NNW framework.

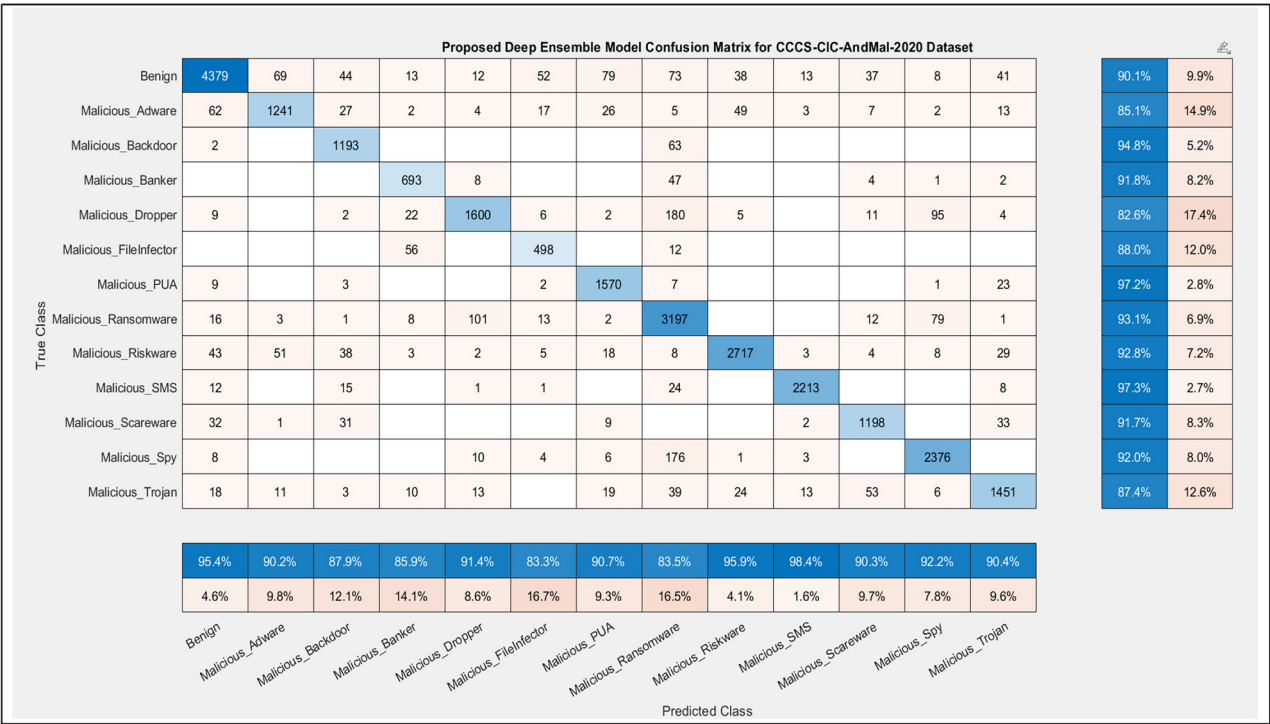


Fig. 12. Confusion matrix of CCCS-CIC-AndMal-2020 dataset using proposed deep NNW framework.

recommended and present approaches, as well as precision, recall, and F1-score, are evaluated. The precision has been good for malware categories including “Riskware” (0.963), “Adware” (0.935), and “Spyware” (0.924), implying that the framework works efficiently in decreasing false positives in those domains. Nevertheless, for certain categories, such as “Backdoor” (0.721) and “PUA” (0.677), it performs ineffectively; in these situations, the precision is lower, reflecting an increased rate of false alarms. The recall for “Riskware” (0.967) and

Malware category	Precision		Recall		F1-Score		Accuracy	
	Proposed	Existing	Proposed	Existing	Proposed	Existing	Proposed	Existing
Adware	0.892	<b>0.935</b>	0.889	<b>0.929</b>	0.891	<b>0.932</b>	85.10	<b>92.82</b>
Backdoor	<b>0.912</b>	0.721	<b>0.948</b>	0.643	<b>0.930</b>	0.680	<b>94.80</b>	59.93
Banker	<b>0.818</b>	0.759	<b>0.917</b>	0.759	<b>0.865</b>	0.759	91.80	<b>92.40</b>
Dropper	0.850	<b>0.850</b>	<b>0.826</b>	0.686	<b>0.838</b>	0.759	<b>82.60</b>	63.96
FileInfector	0.778	<b>0.909</b>	<b>0.880</b>	0.789	0.826	<b>0.845</b>	<b>88.00</b>	70.31
PUA	<b>0.968</b>	0.677	<b>0.972</b>	0.682	<b>0.970</b>	0.679	<b>97.20</b>	69.29
Ransomware	<b>0.920</b>	0.798	0.931	<b>0.944</b>	<b>0.926</b>	0.864	<b>93.10</b>	91.98
Riskware	0.949	<b>0.963</b>	0.939	<b>0.967</b>	0.944	<b>0.965</b>	92.80	<b>96.55</b>
SMS	<b>0.953</b>	0.917	<b>0.973</b>	0.886	<b>0.963</b>	0.901	<b>97.30</b>	93.99
Scareware	0.326	<b>0.836</b>	0.909	<b>0.764</b>	0.480	<b>0.799</b>	91.70	<b>74.32</b>
Spyware	0.000	<b>0.924</b>	0.000	<b>0.835</b>	0.000	<b>0.877</b>	92.00	<b>91.94</b>
Trojan	<b>0.962</b>	0.895	0.894	<b>0.896</b>	0.927	<b>0.896</b>	87.40	<b>89.09</b>
<b>Overall Results</b>	<b>0.799</b>	0.841	<b>0.865</b>	0.813	<b>0.850</b>	0.825	<b>91.40</b>	82.99

**Table 6.** Classwise comparative-analysis between proposed deep NNW and existing approaches<sup>34</sup> for CCCS-CIC-AndMal-2020 dataset. Significant values are in bold.

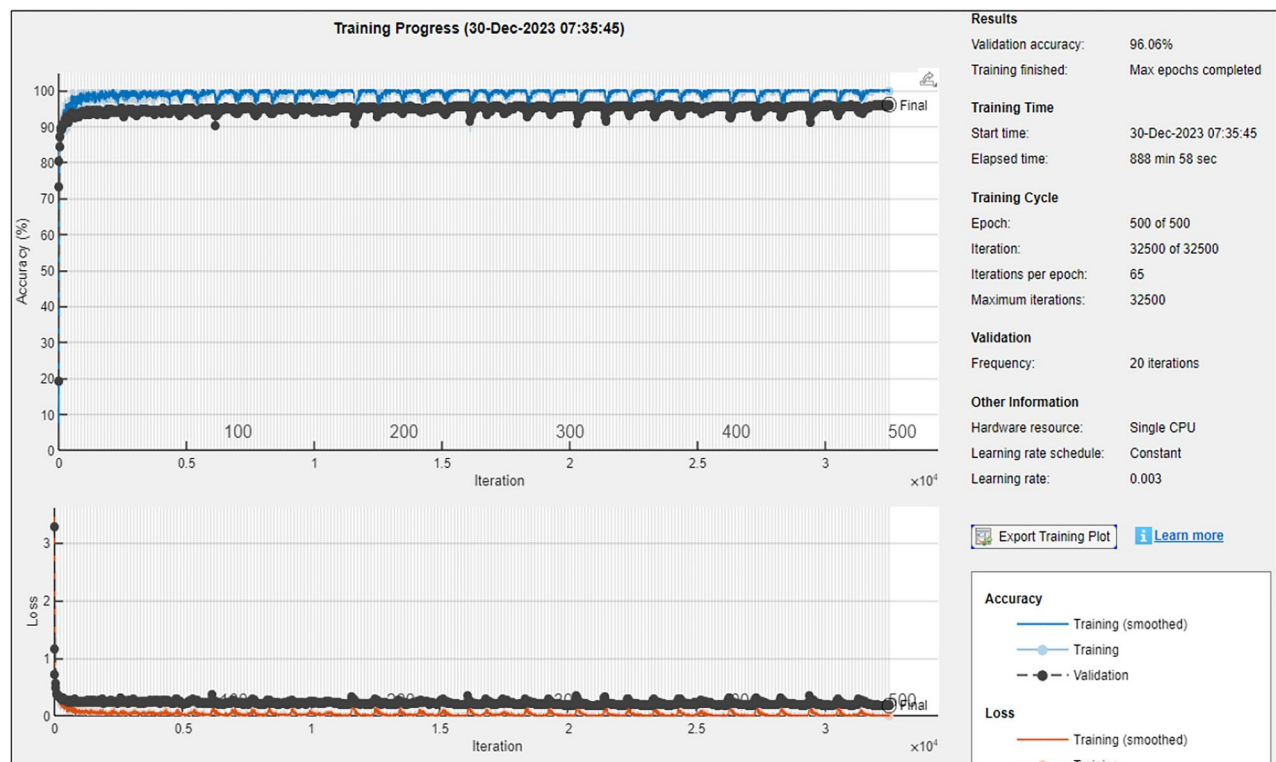
“Ransomware” (0.944) appears to be elevated, revealing that the model effectively identifies a significant number of true positives in these instances. However, the recall values for “Dropper” (0.686) and “Backdoor” (0.643) seem fairly low, revealing that the model has neglected a few real beneficial occurrences. The “Riskware” (0.965) and “Spyware” (0.877) groups have their highest F1-scores, which suggest generic reliability for identifying these kinds of malware. The F1-score balances precision and recall. It is obvious from the lesser F1-scores of categories like “Backdoor” (0.680) and “PUA” (0.679) that it can be tricky to sustain high recall and precision at the same time.

Table 6 depicts that the proposed neural network prevails over the current framework for seven of the twelve malware classes, based on an accuracy analysis. For instance, the proposed solution scores substantially better than the prevailing model in the “Backdoor” category (accuracy of 94.8% vs. just 59.93%) and the “Dropper” class (accuracy of 82.6% vs. 63.96%). Additionally, the recommended approach far surpasses the competition in domains like “FileInfector” (88.0% vs. 70.31%) and “PUA” (97.2% vs. 69.29%). However, the current methodology slightly prevails over the one being suggested for categories like “Adware,” “Riskware,” and “Trojan.” Although the proposed approach delivers exceptionally well in most classes, its overall accuracy (91.40%) is slightly behind the prevailing approach (93.3%), potentially due to its underperformance in several distinct malware classes. However, the recommended technique can identify several complex malware categories, presenting a more trustworthy approach in such scenarios.

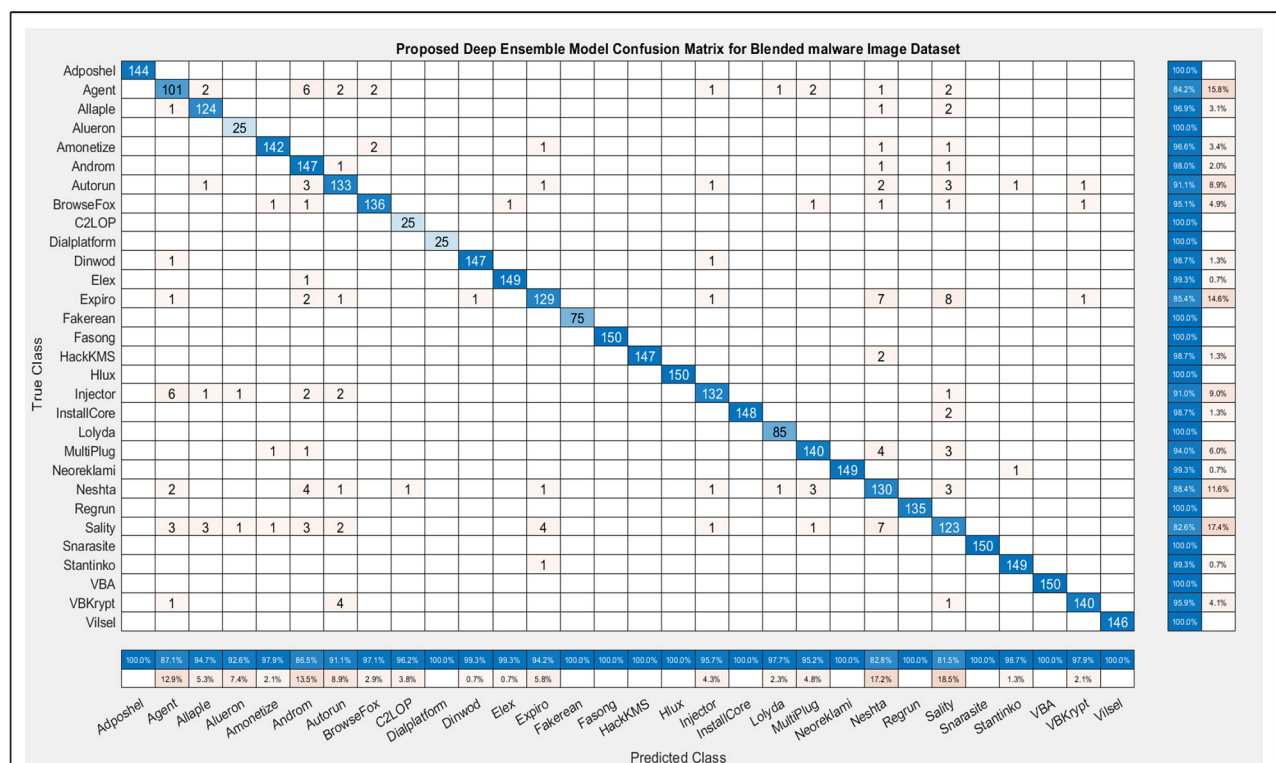
In the consequent exploration, the proposed ensemble deep neural network was employed to process the blended malware image dataset using the ‘Adam’ optimizer and specific hyperparameters, learning rate (0.003), 500 epochs and mini-batch size (150). The findings of the referenced research<sup>67</sup> implies that ResNet50 received the highest precision, recall, and F-score, which was determined at 95% for the blended image dataset. Figures 13 and 14 represent the better performance of the proposed ensemble deep neural network employed for the blended image dataset. The generated results beat the existing benchmark with accuracy, precision, recall, and F-score of 96.06%. These results establish the improved efficiency and reliability of the proposed framework in classifying malware image data. The performance of models is assessed by employing a confusion matrix and training graph as depicted in Figs. 13 and 14 respectively, which serves as the baseline for computing and displaying key metrics such as precision, recall, F1-score, and accuracy for each class. This exhaustive analysis of the proposed technique illustrates significant intuitions into classifier strength to distinguish between individual categories in the blended malware imagery dataset.

The precision, recall, F1-score, and accuracy across multiple malware categories are demonstrated in Table 7. Exceptionally, low false positive rates are evident in categories with high precision values, likewise “Adposhel,” “Amonetize,” and “Androm,” indicating the capacity of the model to determine true positive cases precisely. Lower accuracy and F1-scores for classes like “Fakerean” and “Alueron” could indicate limitations in identifying such categories from others because of common characteristics or complicated data patterns. It is important to emphasize that multiple categories, such as “C2LOP” and “Dialplatform,” have perfect recall but slightly lower precision and F1 scores, implying that although all positive cases were recognized, a substantial percentage of false positives were also present.

The precision for every class is continuously 96.06% in general, reflecting outstanding results. However, the differences in recall, F1 scores, and precision among various malware categories point to potential weaknesses in the framework, notably when it attempts to strike an appropriate balance between recall and precision against specific malware categories. The effectiveness of the framework in the real-life identification of malware tasks, where false positives and negatives might have serious effects, must be adjusted with this balance.



**Fig. 13.** Training result of Blended Image dataset by training against proposed deep NNW framework.



**Fig. 14.** Confusion matrix of Blended Image dataset by training against proposed deep NNW framework.



Class	Precision	Recall	F1-Score	Accuracy
Adposhel	1.0000	1.0000	1.0000	0.9606
Agent	0.7014	0.8417	0.7652	0.9606
Allaple	0.8611	0.9688	0.9118	0.9606
Alueron	0.1736	1.0000	0.2959	0.9606
Amonetize	0.9861	0.9660	0.9759	0.9606
Androm	1.0208	0.9800	1.0000	0.9606
Autorun	0.9236	0.9110	0.9172	0.9606
BrowseFox	0.9444	0.9510	0.9477	0.9606
C2LOP	0.1736	1.0000	0.2959	0.9606
Dialplatform	0.1736	1.0000	0.2959	0.9606
Dinwod	1.0208	0.9866	1.0034	0.9606
Elex	1.0347	0.9933	1.0136	0.9606
Expiro	0.8958	0.8543	0.8746	0.9606
Fakerean	0.5208	1.0000	0.6849	0.9606
Fasong	1.0417	1.0000	1.0204	0.9606
HackKMS	1.0208	0.9866	1.0034	0.9606
Hlux	1.0417	1.0000	1.0204	0.9606
Injector	0.9167	0.9103	0.9135	0.9606
InstallCore	1.0278	0.9867	1.0068	0.9606
Lolyda	0.5903	1.0000	0.7424	0.9606
MultiPlug	0.9722	0.9396	0.9556	0.9606
Neoreklami	1.0347	0.9933	1.0136	0.9606
Neshta	0.9028	0.8844	0.8935	0.9606
Regrun	0.9375	1.0000	0.9677	0.9606
Sality	0.8542	0.8255	0.8396	0.9606
Snarasite	1.0417	1.0000	1.0204	0.9606
Stantinko	1.0347	0.9933	1.0136	0.9606
VBA	1.0417	1.0000	1.0204	0.9606
VBKrypt	0.9722	0.9589	0.9655	0.9606
Vilssel	1.0139	1.0000	1.0069	0.9606

**Table 7.** Class wise parametric analysis against Blended malware image dataset.

### Third phase: application of multimodality by late fusion

The comparative examination of research findings in Table 8 validates the concept that implementing late fusion methods to perform malware classification presents numerous benefits. This table analyzes multiple ensemble models, including RUSBoost, Random Forest, Subspace, AdaBoostM2 and Bagged Trees. These approaches were initially used for two separate datasets: a blended malware image dataset and the CCCS-CIC-AndMal- 2020 quantitative dataset.

The efficiency of every approach is evaluated separately by employing these unimodal datasets. RUSBoost and Random Forest deliver excellent results with accuracies of 94.2% and 94.42%, respectively, for quantitative data, while Subspace performs weakly with an accuracy of 39.70%. The proposed neural network (NNW) excels in models such as Bagged Tree (94.23%) and AdaBoostM2 (87.42%) with an accuracy of 96.06% in the image data set. This initial investigation addresses the efficiency of particular approaches in their respective modalities, e.g., RUSBoost for numeric data and the proposed NNW for imaging data. The results of these frameworks are incorporated between the two datasets through late fusion. The predictions are gathered by applying multiple ensemble techniques, including majority voting, stacking, bagging, and boosting. The category with the highest votes wins the ultimate prediction in the majority vote, which enhances performance. RUSBoost (quantitative) and proposed NNW (imagery) have the highest accuracy rates, at 95.36%. Bagging amplifies efficiency, providing numerous portions of the data to train models and then adding the findings (95.36%). Although they produce slightly less favorable outcomes, stacking and boosting both show the benefits of multimodal integration. Stacking incorporates the findings of different algorithms to train a meta-classifier while boosting concentrates on enhancing the efficiency of weaker classifiers repeatedly.

The best accuracy is obtained through late fusion between the proposed NNW on the image data and RUSBoost on the quantitative data, emphasizing the complementary nature of these models. The proposed NNW effectively handles complex spatial features from visual data, but RUSBoost performs well in addressing unbalanced data and strengthening classification for quantitative data. Through late fusion, the strengths of both models are combined to develop an improved assessment procedure that optimizes classification accuracy. Since late fusion incorporates the strongest features of both modalities, it is the most effective approach. It makes more authentic and precise predictions than any unimodal or ensemble strategy used in separation. Hence, it

Ensemble models	Multimodal implementation (Late fusion)						
	RUS-Boost (%)	Random forest (%)	Subspace (%)	AdaBoost-M2 (%)	BagTree (%)	Proposed NNW (Numeric + Visual) (%)	RUSBoost (Numeric) and proposed NNW (Visual) (%)
Majority Vote	94.34	94.22	55.52	79.34	92.13	94.02	95.36
Stacked Ensemble	74.95	74.38	24.05	53.94	74.83	82.55	82.64
Boosted Ensemble	94.34	76.13	46.84	63.11	73.83	94.02	95.04
Bagged Ensemble	94.34	76.13	50.56	64.68	73.83	94.02	95.36
Existing Mixed Dataset <sup>68</sup>	–	–	–	–	–	92.3	–
Unimodal accuracies							
Numeric Dataset	94.2	94.42	39.70	71.93	89.04	91.40	94.2
Imagery Dataset	93.45	93.35	73.50	87.42	94.23	96.06	–
Existing Numeric Dataset <sup>69</sup>	–	–	–	–	–	93.36	–
Existing Imagery Dataset <sup>70</sup>	–	–	–	–	–	95 precision	–
Existing Pure Text <sup>68</sup>	–	–	–	–	–	96.2	–
Existing Words-changing Text <sup>68</sup>	–	–	–	–	–	87.3	–
Existing Words-missing Text <sup>68</sup>	–	–	–	–	–	92.4	–
Existing Pure Image <sup>68</sup>	–	–	–	–	–	91.2	–
Existing multimodal (API call + Visual) <sup>43</sup>	–	–	–	–	–	–	93.5

**Table 8.** Multimodal versus unimodal comparative analysis for CCCS-CIC-AndMal-2020 and Blended malware image datasets.

is concluded that the results from the late fusion of the proposed NNW on the imaging and numeric datasets are considerably more effective than those from the individual implementations of unimodal or ensemble techniques.

The present multimodal research reveals a notable advancement, obtaining accuracies up to 95.36% with late fusion in numeric and imaging data, different from earlier research that employs a late fusion strategy with an accuracy of 92.3%<sup>68</sup>. This rise in efficiency demonstrates the strength of powerful ensemble techniques that better incorporate both numeric and visual information, such as RUSBoost and the proposed deep ensemble neural network (NNW) method. These strategies use powerful ensemble techniques that boost multimodal information integration, minimize noise in the model, and contribute to generalization. This improvement shows that the late fusion techniques employed in this study outperform conventional approaches by improving fusion across multiple modalities.

The authenticity of the results is robustly endorsed by the comprehensive discussion that summarizes the findings illustrated in Table 6 and Table 7 for unimodal developments, and Table 8 for multimodal occurrence using the late fusion method. The proposed framework demonstrates substantial improvements, specifically in Table 8, where late fusion gains outstanding performance compared with prevailing methods by efficiently amalgamating image and quantitative data. The exhaustive comparative analysis focuses on these developments and interprets them within the framework of existing exploration to highlight their practical and theoretical importance. The use of prominent datasets, stringent experimental standards, and robust evaluation metrics guarantees the authenticity and accuracy of these findings.

Conclusion

The study highlights that a multimodal approach significantly strengthen classification operations, validated by evaluation parameters such as precision, recall, F1-score, and ROC-AUC. Comparative analysis with prevailing approaches endorses the dominance of late fusion performed by the RUSBoost framework (numeric) and proposed ensemble deep neural network (imagery). The proposed multimodal framework effectively mitigates bias in classification outcomes by leveraging the ability of RUSBoost to handle imbalanced data. The late fusion approach demonstrates its superiority by showcasing improved robustness and detecting diverse malware variants. In addition, the results validate the capacity to engage in multimodal analysis to address challenges inextricably linked to malware identification. This study deals with the general concerns of imbalanced data and redundant features by applying SMOTE for data balancing and NCA for feature selection. These outcomes contribute to the advancement of AI-driven malware detection, highlighting the significance of multimodal learning in cybersecurity. The outstanding efficiency of the proposed ensemble deep neural network, specifically in multimodal fusion, designates a promising track for future cybersecurity research. This technique boosts detection rates and improves the robustness of malware classification algorithms, thus facilitating more sophisticated and secure digital environment.

## Future work

Future research could benefit this study by exploring more data modalities, such as network traffic data and behavioral logs, to amplify the adaptability and performance of malware detection systems. Exploring the real-time application of the proposed framework produces significant benefits. For cybersecurity experts, prioritizing explainable AI methods may yield comprehensive insight into the decision-making process of ensemble frameworks, thus amplifying transparency and trustworthiness.

## Computing infrastructure

The computing infrastructure comprises hardware and software requirements for experimental analysis. These are classified as follows: Hardware: A 7 th-generation Core i5 processor with 16 GB of RAM. Software: Windows 11, Matlab 2022a

## Data availability

The datasets analyzed in this study are available at the links provided. <https://www.kaggle.com/datasets/gauravpandharkar/blended-malware-image-dataset><https://www.unb.ca/cic/datasets/andmal2020.html>.

Received: 28 January 2025; Accepted: 26 March 2025

Published online: 23 May 2025

## References

- Pawłowski, M., Wróblewska, A. & Sysko-Romańczuk, S. Effective techniques for multimodal data fusion: A comparative analysis. *Sensors* **23**, <https://doi.org/10.3390/s23052381> (2023).
- Schranko de Oliveira, A. & Sassi, R. J. Chimera: An android malware detection method based on multimodal deep learning and hybrid analysis. <https://doi.org/10.36227/techrxiv.13359767.v1> (2020).
- Schranko De Oliveira, A. & Sassi, R. Hunting android malware using multimodal deep learning and hybrid analysis data (2021).
- Bashir, S., Maqbool, F., Khan, F. H. & Abid, A. S. Hybrid machine learning model for malware analysis in android apps. *Pervasive and Mobile Computing* **97**, 101859. <https://doi.org/10.1016/j.pmcj.2023.101859> (2024).
- Chen, C.-M., Lin, Z.-Y., Ou, Y.-H. & Lin, J.-W. A hybrid malware analysis approach for identifying process-injection malware based on machine learning. *International Journal of Security and Networks* **19**, 20–30. <https://doi.org/10.1504/IJSN.2024.137312> (2024).
- Yu, L., Xu, L. & Jiang, X. A high-performance multimodal deep learning model for detecting minority class sample attacks. *Symmetry* **16**, <https://doi.org/10.3390/sym16010042> (2024).
- Gibert, D., Mateu, C. & Planes, J. Hydra: A multimodal deep learning framework for malware classification. *Computers and Security* **95**, 101873. <https://doi.org/10.1016/j.cose.2020.101873> (2020).
- Arfeen, A., Khan, Z. A., Uddin, R. & Ahsan, U. Toward accurate and intelligent detection of malware. *Concurrency and Computation: Practice and Experience* **34**, e6652. <https://doi.org/10.1002/cpe.6652> (2022).
- Vasani, V. et al. Comprehensive analysis of advanced techniques and vital tools for detecting malware intrusion. *Electronics* **12**, <https://doi.org/10.3390/electronics12204299> (2023).
- Tang, Y. & Chen, S. An automated signature-based approach against polymorphic internet worms. *IEEE Transactions on Parallel and Distributed Systems* **18**, 879–892. <https://doi.org/10.1109/TPDS.2007.1050> (2007).
- Darem, A. A. et al. An adaptive behavioral-based incremental batch learning malware variants detection model using concept drift detection and sequential deep learning. *IEEE Access* **9**, 97180–97196. <https://doi.org/10.1109/ACCESS.2021.3093366> (2021).
- Malware obfuscation techniques. Singh, J. & Singh, J. Challenge of malware analysis. *International Journal of Information Security Science* **7**, 100–110 (2018).
- Elsersy, W. F., Feizollah, A. & Anuar, N. B. The rise of obfuscated android malware and impacts on detection methods. *PeerJ Computer Science* **8**, e907. <https://doi.org/10.7717/peerj-cs.907> (2022).
- Manisha. Volume 8, no. 5, may-june 2017 issn no (2017).
- Ghafur, S. et al. A retrospective impact analysis of the wannacry cyberattack on the nhs. *npj Digital Medicine* **2**, 98, <https://doi.org/10.1038/s41746-019-0161-6> (2019).
- Growley, M. et al. Navigating the solarwinds supply chain attack (2021).
- Saheed, Y. K. & Misra, S. Cps-iot-ppdnn: A new explainable privacy preserving dnn for resilient anomaly detection in cyber-physical systems-enabled iot networks. *Chaos, Solitons & Fractals* **191**, 115939. <https://doi.org/10.1016/j.chaos.2024.115939> (2025).
- Cruickshank, I. J. & Carley, K. M. Analysis of malware communities using multi-modal features. *IEEE Access* **8**, 77435–77448. <https://doi.org/10.1109/ACCESS.2020.2989689> (2020).
- Deng, W. et al. Smart contract vulnerability detection based on deep learning and multimodal decision fusion. *Sensors* **23**, <https://doi.org/10.3390/s23167246> (2023).
- Zhu, D. et al. A transparent and multimodal malware detection method for android apps. In *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM '19*, <https://doi.org/10.1145/3345768.3355915> (ACM, 2019).
- Stahlschmidt, S. R., Ulfenborg, B. & Synnergren, J. Multimodal deep learning for biomedical data fusion: a review. *Briefings in Bioinformatics* **23**, bbab569, <https://doi.org/10.1093/bib/bbab569> (2022).
- Zhao, F., Zhang, C. & Geng, B. Deep multimodal data fusion. *ACM Computing Surveys* <https://doi.org/10.1145/3649447> (2024).
- Guo, W., Wang, J. & Wang, S. Deep multimodal representation learning: A survey. *IEEE Access* **7**, 63373–63394. <https://doi.org/10.1109/ACCESS.2019.2916887> (2019).
- Korthals, T., Hesse, M., Leitner, J., Melnik, A. & Rückert, U. Jointly trained variational autoencoder for multi-modal sensor fusion. In *2019 22th International Conference on Information Fusion (FUSION)*, 1–8, <https://doi.org/10.23919/FUSION43075.2019.9011314> (2019).
- Xue, Z. & Marculescu, R. Dynamic multimodal fusion. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* 2575–2584 (2023).
- Rashinkar, P. & Krushnasamy, V. S. An overview of data fusion techniques. In *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 694–697, <https://doi.org/10.1109/ICIMIA.2017.7975553> (2017).
- Guo, X., Kong, A.-W.-K. & Kot, A. Deep multimodal sequence fusion by regularized expressive representation distillation. *IEEE Transactions on Multimedia* **25**, 2085–2096. <https://doi.org/10.1109/TMM.2022.3142448> (2023).
- Gaw, N., Yousefi, S. & Gahrooei, M. R. Multimodal data fusion for systems improvement: A review. *IIEE Transactions* **54**, 1098–1116. <https://doi.org/10.1080/24725854.2021.1987593> (2021).

29. Al-Azani, S. & El-Alfy, E.-S.M. Early and late fusion of emojis and text to enhance opinion mining. *IEEE Access* **9**, 121031–121045. <https://doi.org/10.1109/access.2021.3108502> (2021).
30. Singh, L., Janghel, R. R. & Sahu, S. P. A hybrid feature fusion strategy for early fusion and majority voting for late fusion towards melanocytic skin lesion detection. *International Journal of Imaging Systems and Technology* **32**, 1231–1250. <https://doi.org/10.1002/ima.22692> (2022).
31. Gao, J., Li, P., Chen, Z. & Zhang, J. A Survey on Deep Learning for Multimodal Data Fusion. *Neural Computation* **32**, 829–864. [https://doi.org/10.1162/neco\\_a\\_01273](https://doi.org/10.1162/neco_a_01273) (2020).
32. Cabral Farias, R., Cohen, J. E. & Comon, P. Exploring multimodal data fusion through joint decompositions with flexible couplings. *IEEE Transactions on Signal Processing* **64**, 4830–4844. <https://doi.org/10.1109/TSP.2016.2576425> (2016).
33. Nemati, S. et al. A hybrid latent space data fusion method for multimodal emotion recognition. *IEEE Access* **7**, 172948–172964. <https://doi.org/10.1109/ACCESS.2019.2955637> (2019).
34. Al-Khater, W. & Al-Madeed, S. Using 3d-vgg-16 and 3d-resnet-18 deep learning models and fabemd techniques in the detection of malware. *Alexandria Engineering Journal* **89**, 39–52. <https://doi.org/10.1016/j.aej.2023.12.061> (2024).
35. Lo, W. W., Yang, X. & Wang, Y. An xception convolutional neural network for malware classification with transfer learning. In *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 1–5. <https://doi.org/10.1109/NTMS.2019.8763852> (2019).
36. Gibert, D., Mateu, C., Planes, J. & Vicens, R. Using convolutional neural networks for classification of malware represented as images. *Journal of Computer Virology and Hacking Techniques* **15**, 15–28. <https://doi.org/10.1007/s11416-018-0323-0> (2019).
37. Nataraj, L., Karthikeyan, S. & Jacob, G. & Manjunath, B. (Visualization and automatic classification, 2011).
38. Tsinganos, N., Fouliras, P., Mavridis, I. & Gritzalis, D. Cse-ars: Deep learning-based late fusion of multimodal information for chat-based social engineering attack recognition. *IEEE Access* **12**, 16072–16088. <https://doi.org/10.1109/ACCESS.2024.3359030> (2024).
39. Ahmad, R. & Alsmadi, I. Data fusion and network intrusion detection systems. *Cluster Computing* **27**, 7493–7519. <https://doi.org/10.1007/s10586-024-04365-y> (2024).
40. Kohout, J., Skarda, C., Shcherbin, K., Kopp, M. & Brabec, J. A framework for comprehensible multi-modal detection of cyber threats. *CoRR* (2021). [arXiv:2111.05764](https://arxiv.org/abs/2111.05764).
41. Qaisar, Z. H. & Li, R. Multimodal information fusion for android malware detection using lazy learning. *Multimedia Tools and Applications* **81**, 12077–12091. <https://doi.org/10.1007/s11042-021-10749-8> (2021).
42. Amrutha, N. & Balagopal, N. Multimodal deep learning method for detection of malware in android using static and dynamic features. *CSI Journal of* **13** (2020).
43. Dai, Y., Li, H., Rong, X., Li, Y. & Zheng, M. M4d: A malware detection method using multimodal features. In Shen, B., Wang, B., Han, J. & Yu, Y. (eds.) *Frontiers in Cyber Security*, 228–238 (Springer Singapore, Singapore, 2019).
44. Amer, E., Zelinka, I. & El-Sappagh, S. A multi-perspective malware detection approach through behavioral fusion of api call sequence. *Computers and Security* **110**, 102449. <https://doi.org/10.1016/j.cose.2021.102449> (2021).
45. Yerima, S. Y. & Sezer, S. Droidfusion: A novel multilevel classifier fusion approach for android malware detection. *IEEE Transactions on Cybernetics* **49**, 453–466. <https://doi.org/10.1109/TCYB.2017.2777960> (2019).
46. Singh, J. et al. Classification and analysis of android malware images using feature fusion technique. *IEEE Access* **9**, 90102–90117. <https://doi.org/10.1109/ACCESS.2021.3090998> (2021).
47. Kumar, R. et al. A multimodal malware detection technique for android iot devices using various features. *IEEE Access* **7**, 64411–64430. <https://doi.org/10.1109/access.2019.2916886> (2019).
48. Chen, S., Lang, B., Liu, H., Chen, Y. & Song, Y. Android malware detection method based on graph attention networks and deep fusion of multimodal features. *Expert Systems with Applications* **237**, 121617. <https://doi.org/10.1016/j.eswa.2023.121617> (2024).
49. Gu, J., Zhu, H., Han, Z., Li, X. & Zhao, J. Gsedroid: Gnn-based android malware detection framework using lightweight semantic embedding. *Computers and Security* **140**, 103807. <https://doi.org/10.1016/j.cose.2024.103807> (2024).
50. Singh, N. & Tripathy, S. Mldroid: Multimodal deep learning based android malware detection. In Muthukumarasamy, V., Sudarsan, S. D. & Shyamasundar, R. K. (eds.) *Information Systems Security*, 159–177 (Springer Nature Switzerland, Cham, 2023).
51. Kim, J., Paik, J.-Y. & Cho, E.-S. Attention-based cross-modal cnn using non-disassembled files for malware classification. *IEEE Access* **11**, 22889–22903. <https://doi.org/10.1109/access.2023.3253770> (2023).
52. Yang, X., Peng, G., Zhang, D., Gao, Y. & Li, C. Powerdetector: Malicious powershell script family classification based on multimodal semantic fusion and deep learning. *China Communications* **20**, 202–224. <https://doi.org/10.23919/JCC.fa.2022-0509.202311> (2023).
53. Ullah, F., Turab, A., Ullah, S., Cacciagran, D. & Zhao, Y. Enhanced network intrusion detection system for internet of things security using multimodal big data representation with transfer learning and game theory. *Sensors* **24**. <https://doi.org/10.3390/s24134152> (2024).
54. Visual, L. & Features. Deep learning fusion for effective malware detection (2024).
55. Trisandhya, P., Vashishtha, S., Hasija, U., Dadhwal, H. & Qureshi, R. Multimodal machine learning approach for detecting spyware and ransomware. In *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)*, 1–8. <https://doi.org/10.1109/ISCS61804.2024.10581161> (2024).
56. Khalifa, M. A., Elsayed, A., Hussien, A. & Hussainy, A. S. Android malware detection and prevention based on deep learning and tweets analysis. In *2024 6th International Conference on Computing and Informatics (ICCI)*, 153–157. <https://doi.org/10.1109/ICCI61671.2024.10485022> (2024).
57. Ayantayo, A. et al. Network intrusion detection using feature fusion with deep learning. *Journal of Big Data* **10**, 167. <https://doi.org/10.1186/s40537-023-00834-0> (2023).
58. Alzubi, O. A., Qiqieh, I. & Alzubi, J. A. Fusion of deep learning based cyberattack detection and classification model for intelligent systems. *Cluster Computing* **26**, 1363–1374. <https://doi.org/10.1007/s10586-022-03686-0> (2023).
59. Kalash, M. et al. Malware classification with deep convolutional neural networks. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 1–5. <https://doi.org/10.1109/NTMS.2018.8328749> (2018).
60. Al-Qadasi, H., Benchadi, D. Y. M., Chehida, S., Fukui, K. & Bensalem, S. Neural network innovations in image-based malware classification: A comparative study. In Barolli, L. (ed.) *Advanced Information Networking and Applications*, 252–265 (Springer Nature Switzerland, Cham, 2024).
61. Keyes, D. S. et al. Entropylyzer: Android malware classification and characterization using entropy analysis of dynamic characteristics. In *2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS)*, 1–12. <https://doi.org/10.1109/RDAAPS48126.2021.9452002> (2021).
62. Rahali, A. et al. Didroid: Android malware classification and characterization using deep image learning. In *Proceedings of the 2020 10th International Conference on Communication and Network Security, ICCNS '20*, 70–82. <https://doi.org/10.1145/3442520.3442522> (Association for Computing Machinery, New York, NY, USA, 2021).
63. Islam, R., Sayed, M. I., Saha, S., Hossain, M. J. & Masud, M. A. Android malware classification using optimum feature selection and ensemble machine learning. *Internet of Things and Cyber-Physical Systems* **3**, 100–111. <https://doi.org/10.1016/j.iotcps.2023.03.001> (2023).
64. Abdulganiyu, O. H., Tchakouch, T. A., Saheed, Y. K. & Ahmed, H. A. Xidintfl-vae: Xgboost-based intrusion detection of imbalance network traffic via class-wise focal loss variational autoencoder. *The Journal of Supercomputing* **81**, 16. <https://doi.org/10.1007/s11227-024-06552-5> (2024).

65. Saheed, Y. K. & Chukwuere, J. E. Xaiensemblel-iov: A new explainable artificial intelligence ensemble transfer learning for zero-day botnet attack detection in the internet of vehicles. *Results in Engineering* **24**, 103171. <https://doi.org/10.1016/j.rineng.2024.103171> (2024).
66. Saheed, Y. K., Usman, A. A., Sukat, F. D. & Abdulrahman, M. A novel hybrid autoencoder and modified particle swarm optimization feature selection for intrusion detection in the internet of things network. *Frontiers in Computer Science* **5**, <https://doi.org/10.3389/fcomp.2023.997159> (2023).
67. Shaik, A., Pendharkar, G., Kumar, S. & Balaji, S. Comparative analysis of imbalanced malware byteplot image classification using transfer learning (2023).
68. Farhan Ullah, S. U. D. C., Ali Turab & Zhao, Y. Enhanced network intrusion detection system for internet of things security using multimodal big data representation with transfer learning and game theory. *Tech. Rep. China*; (2024). <https://doi.org/10.3390/s24134152>.
69. Rahali, A. *et al.* Didroid: Android malware classification and characterization using deep image learning, <https://doi.org/10.1145/3442520.3442522> (2021).
70. Learning. Comparative analysis of imbalanced malware byteplot image classification using transfer (2310).

## Author contributions

Sadia Nazim, Muhammad Mansoor Alam, and Safdar Rizvi contributed equally to the conceptualization and design of the study. Sadia Nazim and Jawahir Che Mustapha wrote the main manuscript text and coordinated the overall project. Muhammad Mansoor Alam, Syed Shujaa Hussain and Safdar Rizvi conducted data analysis and interpretation. Syed Shujaa Hussain and Mazliham Mohd Su'ud prepared figures and supported the manuscript revision process. All authors reviewed and approved the final manuscript.

## Declarations

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to J.C.M.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025