Machine Learning-Driven Approaches for Advanced Collaborative Malware Analysis and Detection

Giragani Nageshwar *[1], Yogesh Rajkumar R [2]
[1] Research Scholar, Department of Computer Science and Engineering, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India
[2] Associate Professor, Department of Information Technology, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India
nageshwargoud29@gmail.com

Abstract. The malware keeps on developing with increasing complexities, and these conventional methods have some challenges. To overcome the challenges of these issues, this research presents a new idea and proposes Hybrid Graph-MaIX, a novel combination of Graph Neural Network and Transformers model designed for advanced collaborative malware analysis and detection. It delves into this approach, explores graph representations concepts for app behavior of application behavior and URL structure jointly, which have the capability to encapsulate and represent the complex relational graphs dependencies among malicious entity units. The transformer updates to improve and enhance contextual understanding and capabilities within learning and empower the model to generalize to various malware families and obfuscation pattern recognition techniques. The proposed framework method will be tested and evaluated on Android apps and URL-based malware datasets, demonstrating its efficacy and applicability as an efficient solution for practical use and showing the effectiveness of the approach in real-world scenarios. By combining the Integrating capabilities of the collaborative intelligence, the proposed system will provide an advanced facility that enables cross-platform and multi-source threat analysis and enhances resilience to emergent attacks. The experimental findings clearly demonstrate that the Experimental results indicate that Hybrid Graph-MaIX achieves an accuracy of the suggested algorithm that exceeds 97.15% outperforming conventional state-of-the-art machine learning and deep learning baselines. This research clearly emphasizes developing scalable and interpretable, highly efficient malware detection through collaborative intelligent capability, which underlines the potential of machine learning collaborative approaches to push state -of-the-art malware detection, yielding a scalable and interpretable high-performance solution for modern cybersecurity ecosystems. The findings will be useful as the results are expected to contribute to the development of next-generation security systems for safeguarding emerging frameworks that protect mobile and web platforms against new environments from emerging threats.

Keywords: Machine Learning-Driven Approaches, Collaborative Malware Analysis, Malware Detection, Hybrid Graph-MaIX, Graph neural networks, Transformer Models, URL malware.

S. P. Vijayaragavan et al. (eds.), *Proceedings of the Global Conference on Sustainable Energy Systems, Smart Electronics and Intelligent Computing (GCSESEIC 2025)*, Advances in Engineering Research 297,
https://doi.org/10.2991/978-94-6239-654-8_53

1. Introduction

Despite these breakthroughs and existing advances, a significant research gap still pervades the area regarding the efficacy and effectiveness of machine learning algorithms and methods for malware analysis and detection on various platforms and techniques for analyzing and detecting malware across diverse platforms in collaborative environments. As an extension to these findings, and to propose my research contribution, which presents further insight, this paper presents Hybrid Graph-MaIX, a graph Neural Network and transformer hybrid approaches as a hybrid model for the next generation method for collaborative malware analysis and detection. Hybrid Graph-MaIX overcomes the limitations of the state-of-the-art approaches by integrating graph-based relational learning with transformation-driven contextual modelling, further hardening against evolving threats. The proposed framework has been applied to Android applications and URL malware detection, demonstrating an accuracy of 97.15%, hence proving its potential as a robust, adaptable solution for next-generation cybersecurity [1]. Malware attacks are increasing significantly and have posed serious threats to individuals, corporations, and governments. Classic detection approaches using either static or dynamic analysis face substantial challenges in detecting new or zero day threat as modern malware uses polymorphic and metamorphic techniques to bypass detection. Although machine learning has enhanced the detection of malware, most of these techniques depend on biased datasets or heavy feature engineering. Research proposes Hybrid Graph-MaIX, a hybrid framework developed by fusing a Graph Neural Network and Transformer module that offers unbiased, efficient, and robust malware detection on diverse datasets [2].

However, malware continues to evolve with more sophisticated packing and obfuscation techniques. As a result, their identification becomes more complex and difficult to identify malware. Traditional systems and traditional ML models have failed been shown to detect sophisticated malware be ineffective at identifying sophisticated. To address these issues, we propose the hybrid Graph-MaIX approach, a hybrid model incorporating a hybrid model namely graph MaIX that combines Graph Neural Network and Transform models. Considering these capabilities makes it possible to identify malware on Android and URLs more effectively and efficiently [3]. The IoT and IoT environments have grown tremendously fast, proliferating at a rapid rate, enabling better connectivity, entertaining tools and healthcare facilities, along with it has come, but at the same time, giving rise to serious threats. The malware botnet attacks involving Mirai have caused attacks on confidentiality, integrity, and availability related to associated with precious data. Traditional approaches have limitations against dynamic threats; hence, there is a need for a more sophisticated design. The proposal made within our research describes Hybrid Graph-MaIX, a hybrid

model incorporating GNN and transformers for making an unbiased and efficient malware detection system [4].

Cloud computing services play a very significant role within and across the private, public, and business sectors, but they remain a continuous target for different zero-day malware and denial of service attacks. Conventional anomaly based techniques possess limitation with regard to limitations regarding zero-day malware detection [14]. In order to overcome some of these limitations, we propose a method named Hybrid Graph-MaIX. It uses a combination of Graph Neural Network and transformers, exploiting not only graph representation but also the sequential characteristic. This solution will lead us to more results in more unbiased and efficient malware detection within the cloud infrastructure [5]. The rate at which malware growth continues remains at an alarming rate, and with most malware employing using obfuscation techniques, they have managed to evade detection. However, signature-based detection works for known malware but is ineffective against zero-day malware, while behavior-based detection and deep learning detection mitigate these challenges. However, no single solution has been capable of effectively identifying all malware [13]. To address these challenges, we proposed the use of Hybrid Graph-MaIX, a hybrid approach that employs Graph Neural Networks and transformers [6]. IoT networks allow interconnection among various devices and services, but face an emerging risk of malware attacks that may affect critical data and cause serious harm. Traditional detection tools are ineffective against ever-developing threats and thus require sophisticated detection tools. To address this issue, we propose a novel approach that presents a new hybrid framework called hybrid Graph-MaIX based on the fusion of graph Neural Networks and transformers [7]. Malware has now become a critical security issue due to rapid technological advancement in the sense that it affects computer systems and stakeholder alike. Usually, naïve users cannot distinguish a malicious application. Thus, the need for intelligent detection arises. Various AI, ML, and DL approaches provide incomplete solutions to combat complex malware and zero-day malware [12]. So, in this context, our contribution proposes a hybrid framework, Hybrid Graph-MaIX, for malware analysis by fusing Graph Neural Network and Transformer models to capture structural and sequential malware behaviors, enabling robust, unbiased, real-time Android and URL environments malware detection [8]. Malware variants are increasingly obfuscated, evading detection by traditional methods.

Many malware variants proliferate rapidly, with a large number of them attempting to bypass traditional signature-based detection methods. Signature-based approaches work only for known threat zero day attacks will still bypass them. Behavior-based and deep learning based approaches work, but mostly partially [11]. In this paper, we propose a hybrid model that learns the structural and sequential representations of Android and URL malware behaviors for unbiased, scalable, real-

time detection [9]. While malware analysis plays a crucial role in the strengthening of cybersecurity, the conventional approaches are disproportionately challenged by the evolution of threats. Here, the work proposes Hybrid Graph-MaIX, a hybrid framework that combines Graph Neural Networks and a Transformer model for capturing both structural and sequential malware behaviors, thus enabling accurate, unbiased, real-time detection in Android and URL platforms, which offers higher performance compared to traditional approaches and could be a scalable solution against zero-day attacks [10].

2. Methodology

2.1 Dataset preparation

The data preparation process begins with the acquisition of two main sources, which are the Android app data and the URL data. Both sources have been populated with benevolent as well as malicious data. This approach ensures that all possible scenarios are covered. The obtained data is further divided into three sets: training, validation, and testing. The stratified sampling technique prevents any ensures that no class of data from getting imbalanced. Once all data is prepared properly, it establishes a foundation for successful feature extraction, model training, and testing of the Hybrid Graph-MaIX framework. After obtaining once all the datasets are obtained and preprocessing them properly, they are split into three sets: training data, test data, and verification data.

2.2 Data preprocessing

The data preprocessing is a step; therefore, it plays an integral role in preparing and enabling the utilization of raw Android and URL malware datasets for use in training purposes. In essence, it first involves preliminary, initial feature extraction based on domain information permissions, API calls, and system attributes for the classification of Android malware, as well as structural attributes for the classification of URLs. To facilitate an equal scale for these extracted attribute are then normalized to ensure that all learning occurs on the equal scale, these extracted attributes are normalized. To maintain data, preserve the integrity of the data, any missing values and duplicate samples are properly addressed and handled accordingly. Although these datasets are considered balanced, it is imperative to note that additional verification was performed to handle procedures are conducted to address existing bias. Its preprocessing step ensures that the Hybrid Graph-MaIX model learns from clean and representative data.

2.3 Graph construction for GNN

The Fig.1 Creating graphs for the GNN component of Hybrid Graph-maIX requires encoding malware data into structured graphs. The node and edge structure will vary depending on the malware as well as the graph. When working with malware on Android, there will be nodes and edges based on calls, permissions, and system

events. Conversely, there will be graphs encoding malware on URLs with different components like tokens, domains, embedded parameters as nodes and edges based on structure and context. By representing malware as graphs, it benefits from the capabilities of graph learning. It detects hidden associations and makes malware detection more precise.

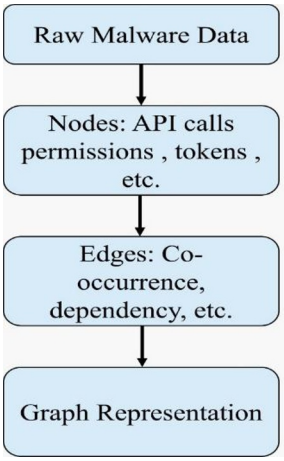

Fig.1. Graph construction process

2.4 Transformer module

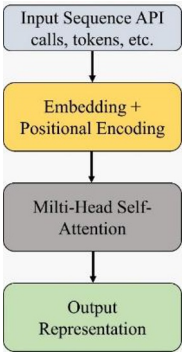

Fig.2. Transformer module workflow

The Fig.2 represents a transformer module within the hybrid GraphMaIX model is designed to pick up on sequential and contextual patterns from malware data that complement the structural insights learned by the GNN. It handles sequences of

API calls, permission requests, and system events in Android datasets and tokenized components such as domain names, path, and query parameters, in URL datasets. Each sequence gets projected into a high-dimensional space and further injected with positional encodings in order for order to be maintained. Using multiple self-attention heads within the multi-head self-attention layers, the transformer learns long-range dependencies and subtle behavioral cues that are indicative of malicious intent. Therefore, it helps the model identify complex attacks across a wide variety of malware types. With this module, Hybrid GraphMaIX leverages the power of attention-based learning to improve the accuracy and robustness of detection.

2.5 Fusion mechanism

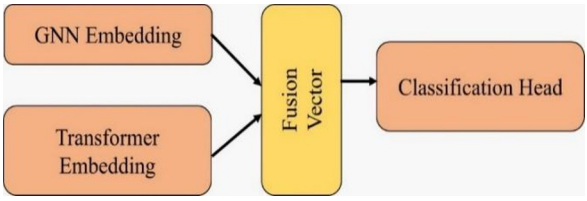

Fig.3. Fusion mechanism combining GNN

Fig.3 represents a fusion mechanism within Hybrid Graph-MaIX that becomes an integral engine that blends the structural understanding derived from the Graph Neural Network with the sequence and contextual information derived from the transformer network. Once these two have extracted their respective latent feature mappings, these are then referenced and concatenated to produce a single feature vector. The resultant vector encodes information about structural associations as well as temporal regularities, which enables the model to make educated decisions. A multi-layer perceptron with dropout follows as the final processing step on the fusion vector, thus imparting it a highly regularized and non-linear representation, which then goes on to predict the malware probability as its classification output.

2.6 Training setup

The training configuration structure for training Hybrid Graph-MaIX will be designed to carefully set up to enhance improve learning and optimize malware detection performance. The model will be trained on a balanced data set with a division made from split for testing and validating. Parameters will be manually set using a technique requiring several steps based on an approach involving various experiments. The Adam optimizer and binary cross-entropy loss function will be employee used to solve the binary classification task and problem. A careful approach involving a cautious strategy combining dropout and weight decay will be incorporated to adapt and avoid model overfitting. Moreover, early stopping based on loss values will be implemented for efficient convergence. All major critical variables are taken into account in the monitoring efforts made for monitoring. All these factors have ensured

optimal generalization capabilities and high reliability for malware detection using tye Hybrid Graph-MaIX.

2.7 Evaluation and ablation

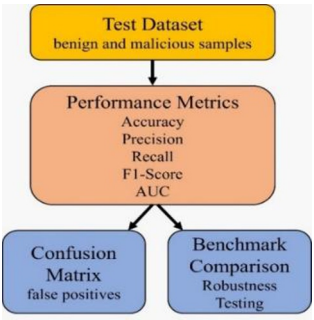

Fig.4. Evaluation workflow

Fig.4 represents the presence of the evaluation stage or phase of Hybrid Graph-MaIX, where in an extensive assessment and analysis of its capability of emphasizing the thorough measurement and examination of its performance of different parameters are carried out after being trained on the malware and benign data, it is then tested on fresh capabilities on various aspects so as to achieve reliability and generality. Once trained, it is validated on new malware as well as benign samples, and various parameters like accuracy, precision, recall, F1-score and AUC are computed and measured to effectively determine malware detection capabilities. Also, the confusion matrices are explored for error classification with special emphasis on false positives and false negatives. Finally comparative analysis with existing state of the art technique is done so as to prove its innovativeness and efficacy. Moreover, performance on obfuscated or adversarial samples can also be ensured as a measure of resilience. All these factors validate that not only does it have a good malware detection rate, but it also stays adaptable with varying malware.

2.8 Deployment

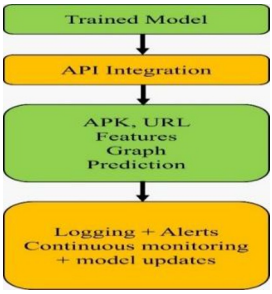

Fig.5. Deployment workflow

The Fig.5 represents that to achieve real-time URL and Android malware detection, the trained HybridGraph-MaIX model is integrated into an interactive interface for easy usage and fast computation. The interface can be implemented as a web portal, mobile app, and browser extension, enabling the user to analyse APS/URL on the fly. Once the data is uploaded, it analyses extracted features, builds graphs, and uses the transformer component for pattern recognition based on sequences. The embeddings obtained are fused and processed in the classification head for prediction. The platform then presents the output with easy-to-interpret labels, for instance, malicious, suspicious and benign, as well as confidence level and additional danger information. Backend and frontend implementation enable efficient processing with low latency and an intuitive user experience, respectively.

3. Result And Discussion

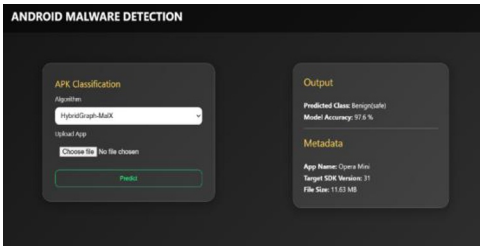

Fig.6. user interface of the hybridgraph-MaIX

The Fig.6 represents the image that presents the user interface of an Android malware detection system that employs the proposed HybridGraph-MaIX algorithm, integrating the Graph Neural HybridGraph algorithm. ing Graph Neural Networks and the Transformer model for advanced malware analysis. On the left, the APK classification section enables a user to upload an Android application. a.apk file and initiate malware prediction using the selected algorithm. Once the prediction has been triggered, the output section shows the classification result in the case; the app has been identified as Benign, together with the model accuracy of 97.6%, which suggests high reliability of the model. Further metadata is provided about the application, including its name, OperaMini, targeting SDK version 31, and having a file size of 11.63 MB. This interface is a good example of how to practically deploy machine learning-driven malware detection by providing a transparent and user-friendly platform for real-time threat assessment in mobile environments.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      9971
           1       0.97      0.97      0.97     10029

    accuracy                           0.97     20000
   macro avg       0.97      0.97      0.97     20000
weighted avg       0.97      0.97      0.97     20000
```

Fig.7. Classification report

Fig.7 represents a further point out the practical implications of ML in malware detection, especially across Android and network platforms, while noting that despite the wide amount of research in the area, there remains a significant gap in understanding how ML performs across diverse platforms in collaborative environments. This proposes a methodical approach, specifically the HybridGraph-MaIX algorithm, which combines graph neural networks and transformers for collaborative malware analysis and detection. It mentions some of the challenges faced in ML-based malware detection, such as imbalanced datasets and adversarial attacks, and emphasizes the need for a robust and adaptable detection system. To provide a high-accuracy solution at 97.15% and suggest future directions for strengthening cybersecurity.

Table 1. Performance evaluation metrics table

Metrics	**Class 0 benign**	**Class1 malicious**	**Average value**	**Weighted score**
Precision	0.97	0.97	0.97	0.97
Recall	0.97	0.97	0.97	0.97
F1-score	0.97	0.97	0.97	0.97
Support	9971	10029	0	20000
Accuracy	0	0	0	0.97

The above Table 1 represents the presence of a classification report that describes the performance of the HybridGraphMaIX model in distinguishing the samples into either between benign and malicious samples. The model achieves an average mean precision, recall, and F1-score of 0.97 for both classes, indicating the correct identification of that it correctly identified 97% of the sample while having minimal false positives and false negatives for any class. This is supported by the balanced nature of the support values, which include 9,971 benign samples and 10,029 malicious samples for a total of 20,000 instances. The overall accuracy of 97% confirms the model's reliability across the entire dataset. The macro and weighted average also stand at 0.97, demonstrating consistency across the two classes in model performance, that is, the model is as good at detecting bening application as it is at detecting malicious

ones. This high-performance balanced across both application types further speaks to the robustness of HybridGraph-MaIX and positions it for real-world malware detection tasks on both the Android and URL platforms.

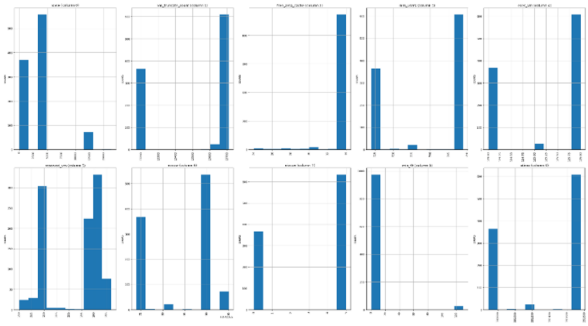

Fig.8. Feature distribution histogram

The Fig.8 represents an image showing a grid of histograms visualising the distribution of values across ten different feature from from a malware-related dataset. Each subplot represents a different column, like state, vm-truncate-count, free-area-cache, and mm-users, among others. The x-axis shows the range of values, while the y-axis displays the frequency. These histograms are very important to understand the statistical behaviour. These histograms are very important to understand the statistical behaviour of each feature, it helps to detect skewed distributions, outliers, and whether a feature should be normalised. The features nvcsw and utime may come from a heavy-tailed distribution, indicating variability in process switch and user time metrics, respectively. The idea of this visualization is to understand what the underlying data charcteristics are like to better understand what the underlying data charateristics are like to better understand where machine learning model performance is coming from and to make the desired preprocessing steps, such as scaling and transformation. This indeed strengthens the reliability and interpretability of malware detection systems like HybridGraph-MaIX.

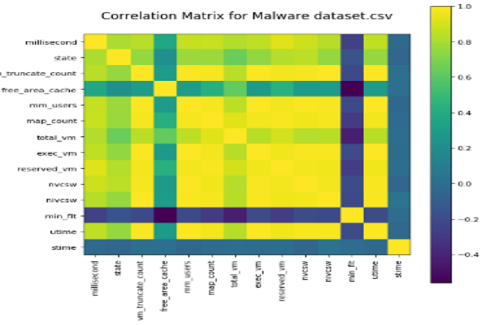

Fig.9. Correlation matrix

Fig.9 represents the importance of malware in protecting digital systemfrom cyber threats, while ML techniques have become increasingly effective in this domain. This is based on a systematic review of the literature conducted from 2020 to 2024, which analysed different ML-based approaches to malware detection. More precisely, it examines types of algorithms used in supervised, unsupervised, and deep learning tools used for data preparation, feature extraction, and model evaluation. Establishes the further identifies the practical implications of ML in malware detection across platforms, such as Android and the network environment. Despite extensive research, one notable gap identified is a lack of effective ML techniques to analyse malware collaboratively across diverse platforms. The research addresses this gap with the proposal of a hybrid model-HybridGraph-MaIX that combines Graph Neural Networks and Transformers, allowing advanced collaborative malware analysis.

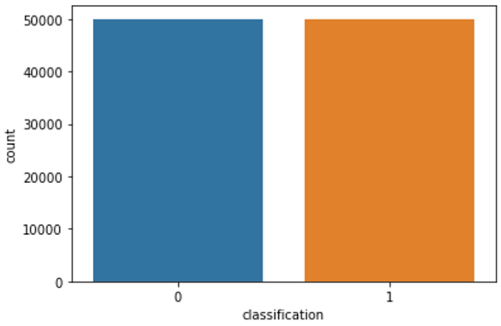

Fig.10. Data count

Fig.10 represents the bar chart showing the class distribution within the malware detection dataset that was used for training and evaluating the HybridGraph-MaIX model. It compares the frequency of two classification categories, usually representing benign class 0 and malicious class 1 samples. Both bars are roughly of the same height, with counts of about 50,000 each; hence, the dataset is well balanced. This balance is important in a binary classification task so that the machine learning model does not get biased toward one class when the samples of that particular class are disproportionately represented. A balanced dataset will improve the reliability of performance metrics like precision and recall and the F1-score, and it contributes to the better generalisation capability of the model over unseen data.

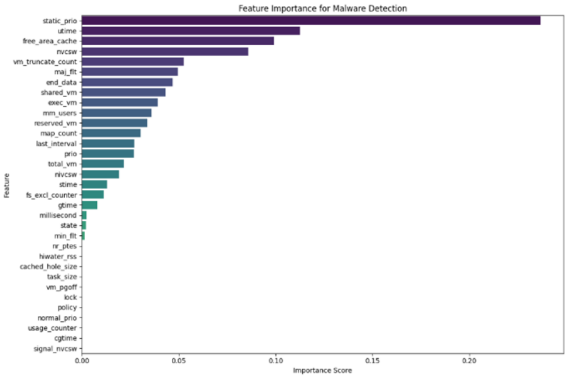

Fig.11. Feature importance score for system attributes

Fig.11 represents the bar chart feature importance for malware detection, plotting the relative contribution of different system-level features used by the hybridgraph-MaIX model in the course of detecting malicious behaviour. Each bar shows a feature, and its length is mapped to its importance score in the model decision making peocess. The top feature appering are static-prio, utime, free-area-cache behaviour, and context switches are among the critical features for differentiating between benign and malicious applications. As many as 35 features are used here, ordered in decreasing order of importance, that provide insights into which attributes most affect malware classification. Similarly, this visualisation allows for feature selection and model interpretability in order to maximize detecction performnace while minimising computational overhead. It further corroborates the robustness of HybridGraph-MaIX by underlining the dependency on meaningful, diverse behavioural indicators on Android and URL-based malware datasets.

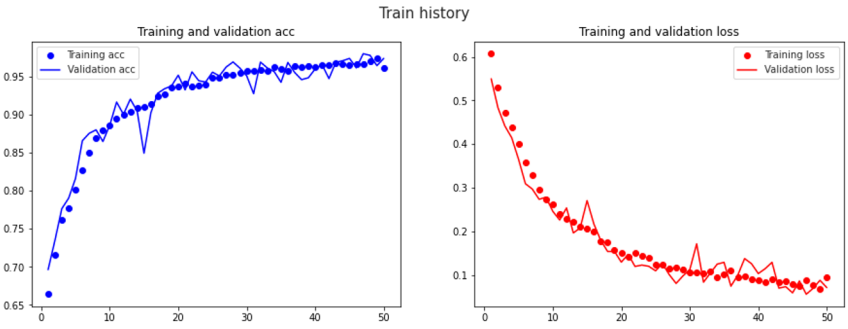

Fig.12. training and validation accuracy plot

Fig.12 represents a training process of the HybridGraph-MaIX model on 50 epochs using two line graphs that analyse accuracy and loss. The graph on the left side depicts an increase in accuracy for both the training and validation sets, which

maintains a 95% accuracy level. The graph on the right side depicts a decrease in loss for both the training and validation sets, which reduces loss significantly from above 0.6 to below 0.1. The near-identity relationship shown in the graph on both sides clearly demonstrates there is very little scope for overfitting. From the above graph, it can be confirmed that HybridGraph-MaIX performs well on malware classification problems on Android and URL platforms.

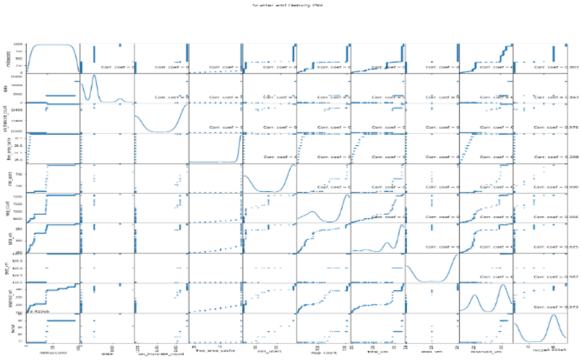

Fig.13. Scatterplot matrix showing feature distribution

Fig.13 represents an image of a scatterplot matrix, better known as a pair plot, that provides a visualisation of the relationships and distributions among multiple features in the malware dataset. Each call along the diagonal shows a histogram for the distribution of one feature pair. Notably, each scatterplot is labelled with a correlation coefficient of zero, indicating no linear relationship between the respective feature pairs. This plot will be very useful for evaluating feature independence and interaction, as it highlights which attributes may contribute uniquely to the model's decision-making process. Most of the correlations are not strong, implying that the features are uncorrelated. This is beneficial in ML models such as HybridGraph-MaIX because it reduces redundancy and enables the model to learn more patterns in the malware detection problem.

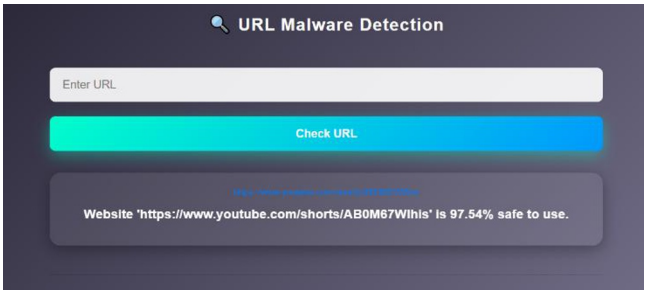

Fig.14. Web-Based interface for real-time URL malware detection

Fig.14 represents image represents an online interface for a URL malware detection system capable of determining the safety level of online resources before visiting them. The online interface contains an input text box marked “Enter URL,” a button marked “Check URL,” and a result display box. AS depicted, the system analyses a given online resource https://www.youtube.com/shorts/AB0M67WIhis and determines it as “97.54% safe to use.” This can be interpreted as the pattern and structure. A system like this plays a critical role in online security as it helps an online user steer clear of malicious online resources. Moreover, it enhances online trust. The above online interface also illustrates the applicability of the HybridGraph-MaIX model for developing an online malware detection platform. It interprets malware detection and represents it as a platform.

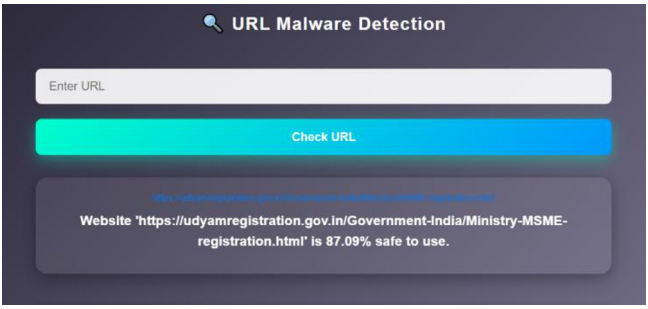

Fig.15. User interface demonstrating URL safety analysis

The Fig.15 represents the image shows an interfacefor a URL malware detection system, which helps detect the safety of URLs before visiting them. The components include an input box marked “Enter URL,” a Check URL button and an output box that shows the safety check result. From the image, it can be seen that it checks a URL http://udyamregisteration.gov.in/Governemnt-India/Ministry-MSME-registeration .html and detects capabilities to analyse the safety and functionality of a loaded URL. This plays a very critical role in cyber security because it helps people quickly identify the safety and functionality of a website before people quckly identify the safety and functionality of a website before preceeding with its usage. It shows how the HybridGraph-MaIX can be put into pratice, it uses malware analysis capabilities and applies them effectively.

Fig.16. URL malware detection interface

Fig 16 represents a picture highlighting an interactive digital display for a cybersecurity solution named URL malware detection. It enables link scanning and testing for safety. A user can enter a link via the given text box and then click on Check URL to start the verification process. Looking at the image, it can be seen that it uses a particular link and, as a result, shows it to be 100.00% unsafe. Solutions like these are highly necessary today for combating phishing attacks, malware attacks, and other cyber threats as they enable a user to check the safety of a link before opening it, particularly in IoT, mobile app, and cloud-based services.

4. Conclusion

In this paper, we have suggested a hybrid model of malware detection known as Hybrid Graph-MaIX to detect malicious Android apps and harmful URLs. The model is a combination of graph neural networks and transformer models. The transformer assists in capturing sequential and contextual patterns, whereas the graph neural network aids in comprehending the structural association of various components. The combination of both lets the model more effectively and accurately identify malware. The suggested system was evaluated using Android and URL samples with benign and malicious cases. Experimental findings indicate that the model had a precision, recall and F1-score of 97.15 and an average of 97.15. The validation and training outcome also indicate that the model is not overfitting, so its performance is constant. This shows that the model can easily generalise to unknown data.Both Hybrid Graph-MaIX are superior to traditional techniques since they do not rely on signatures or manually extracted features to understand both structural and behavioural patterns. It is also less resistant to the effect of obfuscation and zero-day attacks. The user interfaces that were created on APK and URL analysis also indicate that the model can be implemented in realistic systems and in real time.

The model can be expanded in future work to identify various families of malware, make it more efficient in large-scale settings, and improve protection against adversarial attacks. On the whole, this paper shows that a hybrid machine learning solution can be a reliable and scaled model of contemporary malware detection.

REFERENCE

1. A. Almuqren, M. Frikha, A. Albuali and M. Amin Almaiah, "Malware Detection Based on Machine Learning Methods, Analysis, and Tools," *2024 IEEE Eleventh International Conference on Communications and Networking (ComNet)*, Hammamet, Tunisia, pp. 1-11, doi: 10.1109/ComNet64071.2024.10987343, 2024.
2. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran and S. Venkatraman, "Robust Intelligent Malware Detection Using Deep Learning," in *IEEE Access*, vol. 7, pp. 46717-46738, doi: 10.1109/ACCESS.2019.2906934, 2019.
3. H. Alamro, W. Mtouaa, S. Aljameel, A. S. Salama, M. A. Hamza and A. Y. Othman, "Automated Android Malware Detection Using Optimal Ensemble Learning Approach for Cybersecurity," in *IEEE Access*, vol. 11, pp. 72509-72517, doi: 10.1109/ACCESS.2023.3294263, 2023.
4. M. Wazid, A. K. Das, J. J. P. C. Rodrigues, S. Shetty and Y. Park, "IoMT Malware Detection Approaches: Analysis and Research Challenges," in *IEEE Access*, vol. 7, pp. 182459-182476, doi: 10.1109/ACCESS.2019.2960412, 2019.
5. M. R. Watson, N. -u. -h. Shirazi, A. K. Marnerides, A. Mauthe and D. Hutchison, "Malware Detection in Cloud Computing Infrastructures," in *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 192-205, 1 March-April 2016, doi: 10.1109/TDSC.2015.2457918.2016.
6. O. A. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," in *IEEE Access*, vol. 8, pp. 6249-6271, doi: 10.1109/ACCESS.2019.2963724, 2020.
7. Balaji, A., Sathyasri, B., S, V.V.R., Indumathy, D., Krishnan, R., Vanaja, S.: Intruder Alert System in Smart Home based on IoT Technique. (2022). https://doi.org/10.1109/icpects56089.2022.10047243.
8. M. J. Hossain Faruk *et al*., "Malware Detection and Prevention using Artificial Intelligence Techniques," *2021 IEEE International Conference on Big Data (Big Data)*, Orlando, FL, USA, pp. 5369-5377, doi: 10.1109/BigData52589.2021.9671434, 2021.
9. A. Almuqren, M. Frikha, A. Albuali and M. Amin Almaiah, "Malware Detection Based on Machine Learning Methods, Analysis, and Tools," *2024 IEEE Eleventh International Conference on Communications and Networking (ComNet)*, Hammamet, Tunisia, pp. 1-11, doi: 10.1109/ComNet64071.2024.10987343, 2024.
10. Thakur, P., Kansal, V. & Rishiwal, V. Hybrid Deep Learning Approach Based on LSTM and CNN for Malware Detection. *Wireless Pers Commun* 136, 1879–1901 https://doi.org/10.1007/s11277-024-11366-y(2024).
11. SP Vijayaragavan, B.Karthik, TVU Kiran Kumar," A DFIG based wind generation system with unbalanced stator and grid condition". 2014, Middle-East Journal of Scientific Research, vol-20,DOI: 10.5829/idosi.mejsr.20.08.11384.2014.

12. Karthik, B., Krishna Kumar, T., Vijayaragavan, S.P. *et al.* RETRACTED ARTICLE: Removal of high density salt and pepper noise in color image through modified cascaded filter. *J Ambient Intell Human Comput* 12, 3901–3908 https://doi.org/10.1007/s12652-020-01737-1. (2021).
13. Vijayaragavan, S. P., B. Karthik, and T. V. U. Kiran Kumar. "Effective routing technique based on decision logic for open faults in fpgas interconnects." *Middle--East Journal of Scientific Research* 20 808-811(2014):.
14. A. Ranjith, S. P. Vijayaragavan, N. V and N. Muthukumaran, "An IoT based Monitoring System to Detect Animal in the Railway Track using Deep Learning Neural Network," *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, pp. 1246-1253, doi: 10.1109/ICESC54411.2022.9885303, 2022.