

MALWARE OPTIMIZATION DETECTION USING VARIOUS ADVANCED THREATS FOR TOXIC COMMENT ANALYSIS

Ms. R. Akshaya¹, Department of Computing Electrical and Electronics Engineering, university
College of Engineering, Nagercoil

Ms. U. Sivakumari², Department of Computing Electrical and Electronics Engineering, university
College of Engineering, Nagercoil

Dr. C. Mythili³, Assistant Professor and Head, Department of Electrical and Electronics Engineering, University
College of Engineering Nagercoil, Tamil Nadu, India

Abstract:

The Malware optimization detection using various advanced threads of toxic comment analysis using an algorithm, namely Double Deep Q-Network (DDQN) with Bi-LSTM of cybersecurity in malware with NLP of toxic comments. With increasing cyber threats and harmful online interfaces, there are dual challenges for the digital ecosystem. The malware infiltration and toxic communication often coexist to compromise trust, privacy, and user safety. The traditional detection model treats these domains in separate fields. The main result is to limit against rapid evolving attacks. This study proposed a hybrid work of Double Deep Q-Networks (DDQN) for optimized malware detection and Bi-directional Long Short-Term Memory (Bi-LSTM) for toxic comment analysis. The integration with the decision-making system. The DDQN enhanced threat prediction through reinforcement learning by dynamically optimizing detection strategies of polymorphic and zero-day malware. The bi-LSTM module captures contextual dependencies in online text that enable robust identification of subtle, disguised, and adversarial toxic expressions. The practical evaluation demonstrates that the combined model achieved higher precision and adaptability compared to standalone approaches. The offering of a scalable defense mechanism across heterogeneous platforms. The malware intrusions and toxic interactions by fostering a secure and trustworthy cyber environment. The overall hybrid accuracy of 93.5 % of average across domains and has higher robustness than individual models.

Keywords: Malware detection, Toxic comment analysis, Double deep Q-Network (DDQN), Bi-directional Long Short-term Memory (Bi-LSTM), Cybersecurity, Online safety

1. Introduction

The IoMT malware detection approaches of analysis and research challenges of an advancement in information and communication technology have changed the entire landscape of computing. It becomes essential to protect the IOT environment from malware attacks. There is also a dark side to

this suffers from various security and privacy issues [1]. The combating of online malicious behavior by integrating harmful news and toxic comments. The effective methods for detecting and categorizing malicious content are proposed and discuss the highlights of the differences between the approaches of combating malicious behavior. It provides a valuable insight into the practical identification and categorization strategies as

they address the pressing challenges of our digital society [2]. The swarm optimization and ML applied to PE malware detection towards cyber threat intelligence. The cyber threat intelligence, such as analysis of application and their metadata for potential threats. The hypothesis was that the common sections, excluding text and data, such as the impact on malware determination [3]. The survey of malware analysis using community detection has considerably increased the time required to detect malware. This issue is worsened by the spread of many variants of the same malware. Dealing with this issue can be leveraged to achieve bulk detection of malware variants to identify malicious communities instead of focusing on the detection time [4].

The survey on adversarial attacks for malware analysis aims to provide an encyclopedic overview of adversarial evasion attacks specifically targeting malware detection and classification systems, standing apart from previous surveys by focusing exclusively and comprehensively on this unique application domain. We identify open problems and propose future research directions for developing more practical, robust, efficient, and generalized adversarial attacks on malware classifiers [5]. The stacked Bi-LSTM for Advanced Toxicity detection in the comment classification of this exposed user to the risk of harassment. The efforts to address toxic comments in online forums have faced challenges with current solutions [6].

To detect abusive comments using ensemble deep learning algorithms. The malware analysis using artificial intelligence technologies facilitated our way on the internet and provided us with great liberty.

The problem has led to building better models for classifying the abusive comments [7]. Abusive comment identification on Indonesian media data using hybrid deep learning. This is a serious problem that must be controlled because the act has an impact on the victim's psychology and causes trauma, resulting in depression. The implementation of this method on a large amount of data to see if the method is able to produce good performance on data that has much more capacity [8]. Detecting toxic comments using the model can provide a platform to share news, information, and social interaction. The social media platform is using them to harass and threaten others in such a resulting in cyberbullying. Toxic comments are online remarks that are insulting, abusive, and inappropriate, and frequently cause other users to quit a debate[9]. The comparison of ML techniques for text analytics to detect the severity of hate comments online. The approach to combat harassment in online platforms is by detecting the severity of abusive comments that have not been addressed [10].

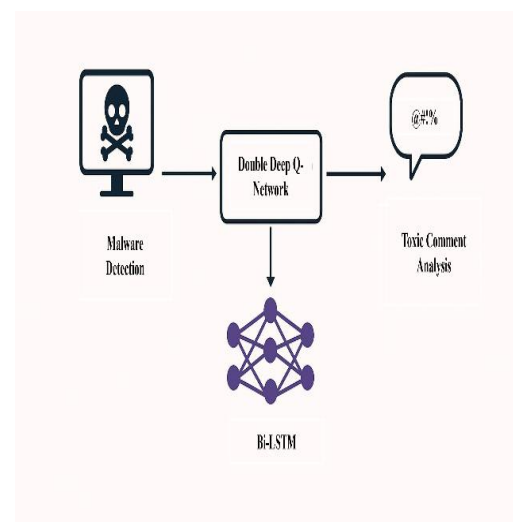


Figure 1: Malware optimization detection using an advanced thread of toxic comment analysis

The user leaves countless comments on various social media platforms, news portals, and forums. The remarks on harmful comment classification of future research, of research gap, and recommendations [11]. The combating of online malicious behavior by integrating ML and Deep learning methods for harmful news and toxic comments. The identification and recognition of harmful news and toxic comments aim to counteract the detrimental impact on public perception [12]. An effective method for detecting the categorizing malicious content is proposed and discussed. It provides valuable insight into the practical identification and categorization of harmful news and toxic comments, highlighting unique facts of these advanced computational strategies as they address the pressing challenges of our digital society [13]. The survey of malware analysis using community detection of fast proliferation of different types of malware targeting organizations and individuals, consistency increasing the time required to detect malware. To deal with these issues, the technique used to detect a leveraged approach to achieve bulk detection of malware families and identify malicious communities instead of focusing on the detection of time [14]. The misuse by spammers, haters, and trolls makes costly content moderation necessary **Figure 1**. The concept of toxicity and character in real-world applications, such as semi-automated comment moderation and troll detection[15].

2. Literature Review

In their 2020 study, Ravinder Ahuja, Alisha Banga, S.S.C. Sharma, et al. [7], presented a comprehensive review of how we can share, like, and comment on any post on social media, but this liberty has caused a severe

threat to humans. Unfortunately, the online interaction among users with such ease involves harassment, abuse, and bullying actions. The problem has led to building up better model for classifying the abusive comments. We found that the CNN, LSTM blend with fast text word embedding performs better than all the accuracy.

In their 2023 study, Alaa Marshan, Farah Nasreen Mohamed Nizar, Athina Loannou, and Konstantina Spanaki, et al. [10], presented a comprehensive review comparing ML and deep learning techniques for text analysis of detecting the severity of hate comments online. This study develops an efficient model to detect the severity of abusive language in online platforms that offers implications both to theory and practice. The effect of text pre-processing on the performance of the machine and deep learning model and word embedding for the model. The development of an effective model to detect the severity of abusive language in online platforms offers important implications.

3. Methodology

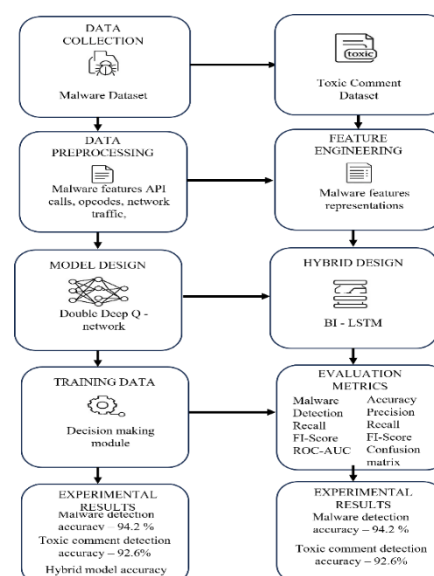


Figure 2: Malware optimization detection using DDQN and Bi-LSTM

The above **Figure 2** represents the methodology of malware optimization detection using DDQN and Bi-LSTM for Toxic Comment Analysis. The malware optimization detection using various advanced threads of toxic comment analysis by using a Double Deep Q-Network, stands for DDQN with Bi-LSTM.

3.1 Data gathering

The data collection in the malware dataset includes datasets such as malware bytes, Virus Share, and Drebin of Android malware containing static and dynamic features. The Toxic comment dataset used publicly available datasets like Kaggle Jigsaw toxic comment classification and Wikipedia talk page comments. The data collection stages of the research of the two major branches. The malware data includes such information as malware data to provide both static features of code structure and permission, and API calls. The dynamic feature of the runtime behavior system calls of network activity. The data contains a large volume of text data, namely for categories like toxicity to hate speech and threats, and insults **Figure 3**.

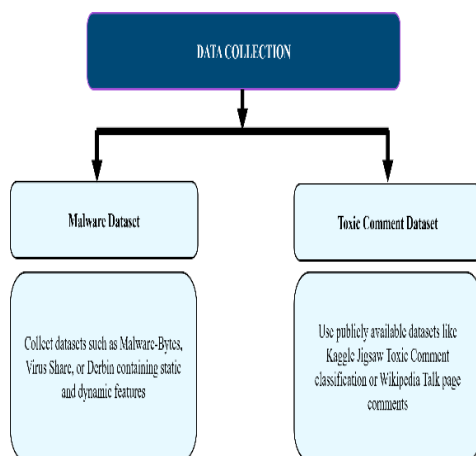


Figure 3: Data collection process for malware and toxic comment data

3.2 Data preprocessing

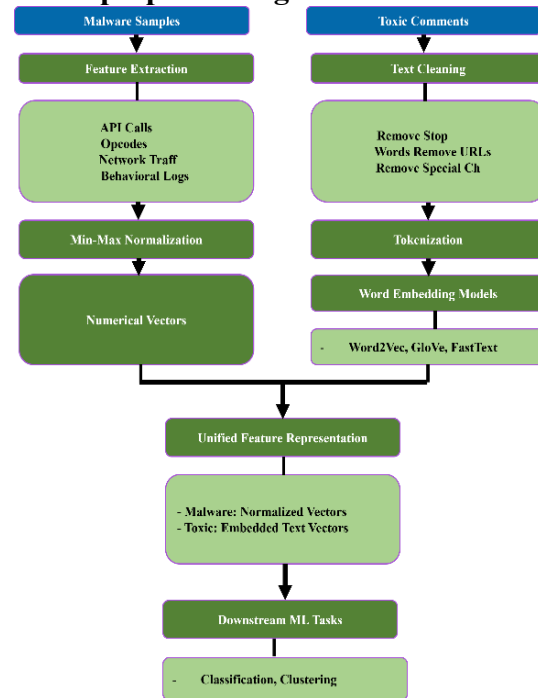


Figure 4: Data preprocessing for malware feature and toxic comment features

The following equation (1) and (2) shows a method of data preprocessing in a normalization of the minimum and maximum value of scaling X is equal to the raw feature, and x' is equal to the normalized value. The word embedding of words of GloVe and FastText, where each word w_i is mapped to a dense vector dimension d .

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

$$w_i \in R^d \quad (2)$$

The malware feature to extract API calls, Opcodes, network traffic, and behavioral logs. The normalized numerical feature of min-max scaling is used in the data preprocessing. The text feature of the toxic comments, such as clean text with removed stop words. By using URLs and special characters that are used to apply word embeddings of word2Vec of

GloVe, and fasttext for vector representation in the following **Figure 4**.

3.3 Feature engineering

The following equations (3) and (4) malware state representation of binary feature, where s_t is the malware state at time t . Defined by API calls, Opcodes, and network logs. The Bi-LSTM input data of text sequence encoding, where X is the embedded toxic comment sequence.

$$S_T = \{f_1, f_2, f_3, \dots, f_n\} \quad f_i \in \{0, 1\} \quad (3)$$

$$X = \{f_1 - f_2 - f_3, \dots, f_n\}, \quad x_t \in R^d \quad (4)$$

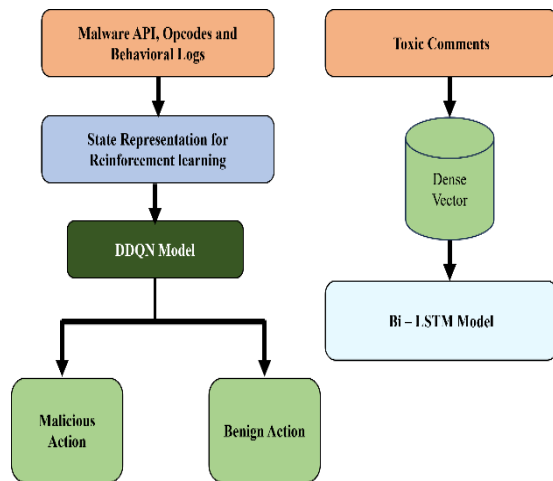


Figure 5: Dual stream feature engineering for malware detection

The malware side generates state representations for reinforcement learning of binary features for malicious action. The toxic comment side encodes the text sequence into dense vectors for the Bi-LSTM input. The extracted malware feature of API calls, opcodes, and behavioral logs is converted into a state representation. The reinforcement learning DDQN can process the data. The states typically represent whether action and behavior are malicious and benign in binary and multi-class form. The enabling agent learn

optimal detection strategies. The toxic comment side can be cleaned textual data is converted into a dense vector. The embeddings of using word2Vec, GloVe, and fastText. The vector preserves the semantic and contextual meaning of text, making them as suitable input data for the Bi LSTM model. The above **Figure 5** indicates that the captures sequential dependencies in toxic and toxic comments.

3.4 Model design

The following equation (5) represents the DDQN for malware detection of the Q-value update rule of DDQN. Where s shows the current state, a shows the action, and r shows the reward. Γ shows the discount factor, and α shows the learning rate. Θ shows online network weight, and θ shows target network weight.

$$Q_{(s,a;\theta)} \leftarrow Q_{(s,a;\theta)} + \alpha[r + \gamma Q(s', \argmax_{a'} Q(s', a'; \theta^-); \theta) - Q(s, a; \theta)] \quad (5)$$

The Double Deep Q-Network stands for DDQN is used for malware detection and optimization. The DDQN learns through state-action reward feedback, dynamically updating detection strategies for polymorphic zero-day malware. By using a Bi-LSTM using the process text is used in both forward and backward directions to capture the contextual meaning of toxic comments. The detection of subtle and disguised toxicity. The scalability of the **Figure 6** module as a meaning it can be extended to additional domains of phishing detection and spam filtering without redesigning the entire system. Join the framework activity in both models to feed their output results into a decision fusion layer. The allowing system acts as a multiple-domain detector of a hybrid structure,

ensuring that the system protects against technical threats and malware simultaneously.

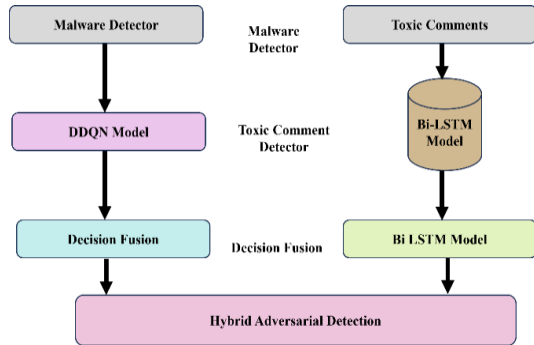


Figure 6: Hybrid adversarial detection work utilizing DDQN and Bi-LSTM for multi-domain threat protection

3.5 Hybrid integration

$$F_{(x)} = \arg \max(w_1 PDDQN(c | x) + w_2 \cdot Bi - LSTM(c | x)) \quad (6)$$

The following equation shows that (6), (7), (8),(9) in Bi-LSTM toxic comment classification using forward LSTM, Backward LSTM, Bi-LSTM hidden state combination, and classification layer softmax.

$$h_t = (W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (7)$$

$$h_t = (W_{xh}x_t + W_{hh}h_{t+1} + b_h) \quad (8)$$

$$h_t = [h_t; h_t] \quad (9)$$

$$\hat{y} = \text{softmax}(W_h h_t + b) \quad (10)$$

The malware detection result as an output data in DDQN data, and the toxic comment classification output of Bi-LSTM are fused in a decision-making module. The weight voting and ensemble mechanism determines the final classification of malware, toxic malware only, and toxic only in a safe. The hybrid integration stage of the layer acts as the central intelligent system that combines the output results from both DDQN and the wall detection data. The Bi-LSTM of toxic comment detection using a decision fusion of an instance instead of

relying on a single model of the system applies weight voting. The following **Figure 7** indicates a representation of the ensemble mechanism that balances aware data and the toxicity production.

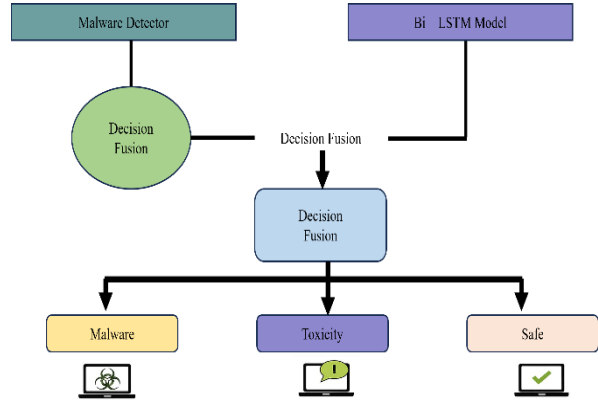


Figure 7: Hybrid decision fusion

Malware using Toxic data represents a device by uses a system in a compromised state and communicates harmful data. The Malware only uses a Technical threat, but no harmful communication. The Toxic is only used for malware detection, but the toxicity language is present. The safe data is used to clean from both fettered data from both the malware toxicity. Resilience data used to dual-layer detection increases robustness against multiple vectors of cyber attack, including malware spreading through toxic and phishing. The optimization ensures the weight can be dynamically turned based on real-time feedback. The improvement in adaptability across the different platforms and datasets.

3.6 Training and optimization

$$L = -\sum_{i=1}^N y_i \log(\hat{y}_i) \quad (11)$$

$$h' = h \odot \text{Bernoulli}(p) \quad (12)$$

The training data and optimisation data are used to train the model DDQN using a dataset with a reward function. The correct detection

4. Future scope

```

graph TD
    A["Model Evolution & Adaptability  
Soft Supervised Learning"] --- B["Multinomial Threat Detection  
Cross-Model Fusion"]
    A --- C["Ethical AI & Governance  
Explainable AI"]
    B --- D["Real-Time Ecosystem Integration  
Edge AI"]
    C --- E["Cross-Domain Applications  
Educational Platforms"]
    D --- E
    C --- B
    A --- D
  
```

Model Evolution & Adaptability
Soft Supervised Learning

Multinomial Threat Detection
Cross-Model Fusion

Ethical AI & Governance
Explainable AI

Real-Time Ecosystem Integration
Edge AI

Cross-Domain Applications
Educational Platforms

4.1 Model Evolution and Adaptability

The self-supervised learning method integrates self-supervised techniques to reduce dependency on labelled data and improve adaptability to emerging threats. The federated learning deployments indicate a

decentralised model training across user devices to enhance privacy and reduce central data exposure. The zero-day threat simulation denotes the development of synthetic data pipelines to simulate unseen malware and linguistic toxicity for proactive model training.

4.2 Multimodal threat detection

The cross-modal fusion indicates an extended detection beyond text and binaries to include images, videos, and voice data using multimodal deep learning. The behavioural biometrics indicate an incorporated user interaction pattern of typing rhythm, navigation flow to detect anomalous behaviour indicative of cyberbullying and malware

4.3 Ethical AI and governance

The explainable AI indicates an embedded interpretability module to justify detection decisions, fostering transparency and trust among users and regulators. The bias auditing framework indicates continuous monitoring for demographic and linguistic bias in toxic comment detection to ensure equitable moderation. The privacy compliance engines indicate an alignment detection system with evolving global privacy laws of GDPR, DPDPA, through automated compliance checks.

4.4 Real-time ecosystem integration

The edge AI optimisation indicates a deployment of a lightweight version of the model on mobile and IOT devices for real-time, low-latency threat detection. The social graph analysis indicates an integration with network topology models to detect coordinated cyberbullying and malware propagation across user clusters. The alert prioritisation system denotes the use of

reinforcement learning of a DQN to rank and route alerts based on severity, context and user vulnerability.

4.5 Cross-domain applications

The educational platforms indicate an adaptive toxic comment detection for online learning environments to protect students and educators. The healthcare social networks denote an application of malware and toxicity filters to protect patient communities and telemedicine platforms. The smart cities and public forms show an extension of the framework to civic engagement platforms where misinformation and abuse can undermine public trust.

4.6 Self-supervised learning

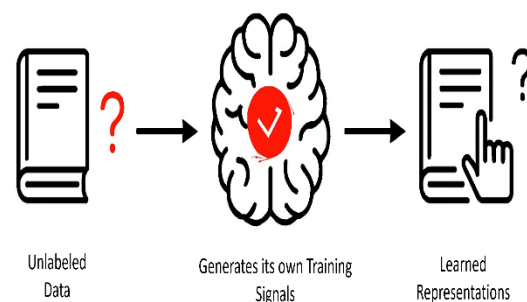


Figure 10: Self-supervised learning architecture

Self-supervised learning is a powerful technique where the model trains itself using patterns found in raw and unlabeled data. Instead of relying on manually annotated shows an example it creates its own learning tasks, including predicting missing sequences and reconstructing corrupted inputs. The malware detection is used to help the model understand hidden behaviours by analysing API patterns Figure 10. In toxic comment

analysis and it enables the system to grasp subtle language cues and relationships without human labelling. This makes the frame highly scalable and adaptive to new threats with minimal manual effort.

4.7 Unlabeled data



Figure 11: unlabeled data architecture

The above Figure 11 representation of a book and a red question mark is also known as the unlabeled data. This is a raw input, such as malware code and social media comments, and that has not been manually labelled. The model does not know what is toxic and malware malicious yet. But it can still learn from patterns within the data. Figure 10 represents raw and unannotated input data, such as malware binaries and social media comments. Symbolising the absence of a label and predefined categories. The unlabeled data below the icon reinforces the idea that the model must learn patterns without human guidance.

4.8 Generates its own training signals

The model creates interval task of an predicting missing API calls and masked words to learn patterns. The learned representations indicates an self generated signals help the model build robust features for classification and detection. Instead of

relying on external labels of the model creates an internal task to teach itself. The malware might predict missing API calls. The toxic comments might have masked word and sentence structure. This stage is where the model begins to understand relationships and behaviours without human supervision.

4.9 Learned representations

After training on unlabeled data using self generated task and the model develops an internal representation, including the feature map and semantic embeddings that capture meaningful patterns. These representation helps the system understand structure, behaviour and the context in new inputs. For example, it can recognise toxic language patterns or malware signatures without needing explicit labels. These learned features are then used in downstream tasks like cyberbullying detection and malware classification, enabling accurate and privacy-preserving threat identification.

4.10 Privacy-preserving threat detection

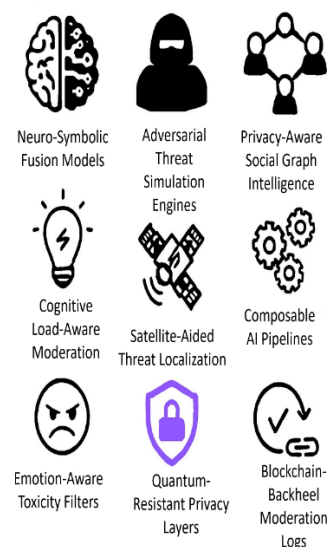


Figure 12: Privacy-preserving threat detection

The above Figure 12 next generation innovation in privacy-preserving threat

detection. This section explores emerging technologies that push the boundaries of security. The adaptive threat detection on social media platforms. The introduction of neuro neuro-symbolic model combines deep learning with logical reasoning for explainable decisions. The adversarial simulation engines that train models against synthetic threats. The privacy-aware graph intelligence helps to detect coordinated abuse without exposing user data. While emotion-aware filters capture subtle toxicity beyond keywords. The innovations, such as blockchain and backed moderation logs, and quantum-resistant encryption, ensure long-term trust and resilience. These trends to redefining how platforms can protect users while respecting privacy and ethical standards.

4.11 Neuro-symbolic fusion model

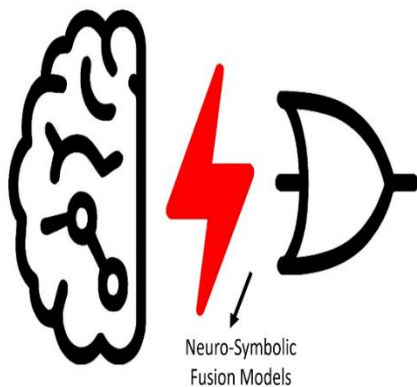


Figure 13: Neuro-symbolic fusion models

The neuro-symbolic fusion combines the pattern recognition power of deep learning with the clarity of symbolic reasoning. The neural network detects complex, hidden patterns in data. While symbolic logic applies rule-based reasoning to interpret behaviour, including toxic language and malware actions. This hybrid reasoning is used to interpret behaviour, including toxic language and

malware actions. This hybrid approach enables explainable, context-aware threat detection, making it especially valuable for platforms that require transparency, legal accountability and ethical governance Figure 13.

4.12 Adversarial threat simulation engines

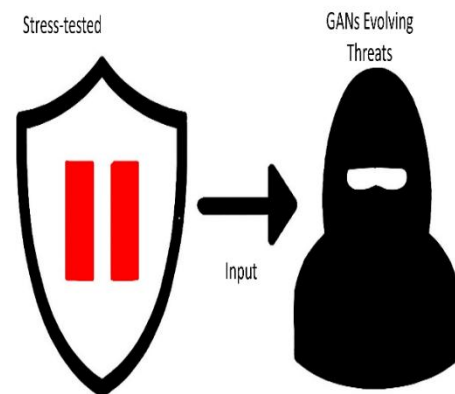


Figure 14: Privacy-preserving detection

The above Figure 14 adversarial threat simulation engines are designed to proactively test and strengthen the threat detection system by generating synthetic and evolving threats. These engines use generative Adversarial Networks (GANs), a type of AI that pits two models against each other. One generates fake data of the adversary, and the other tries to detect it.

The social media safety for malware detection using GAN can simulate new variants of malicious code that mimic real-world attacks but are slightly altered to evade detection. For toxic comment analysis using GANs, adversarial text including sarcastic, coded and manipulated language that challenges the model's ability to flag harmful content. By training on these synthetic adversaries, your system becomes more resilient to zero-day threats of previously unseen malware and abuse patterns. The language of an intentional manipulation of intentional misspellings, slang and context

shifting. This approach helps to build robust, future-proof models that can anticipate and neutralise threats before they cause harm and making it a critical innovation for privacy-preserving real-time moderation.

4.13 Privacy Aware social graph intelligence

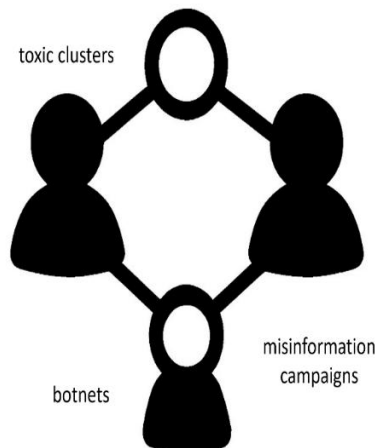


Figure 15: privacy-aware social graph intelligent

The privacy-aware social graph intelligence approach uses a graph neural network of GNN. The model user interactions as dynamic networks enabling detection of coordinated abuse, botnets and misinformation campaigns. By analysing patterns in connections, including reposts, mentions and shared links to the system identifies toxic clusters and malware propagation paths. Crucially, it does so without exposing individual user identities, ensuring privacy is preserved while maintaining platform safety and integrity Figure 15.

4.14 Cognitive load aware moderation

This approach applies behavioural psychology to improve how and when moderation alerts are delivered to users Figure 16. Instead of flagging every potential threat immediately, the system uses the user fatigue

models to estimate cognitive load and how mentally taxed a user might be at any moment. By throttling alerts based on this load. The system avoids overwhelming users with constant warnings and false positives.

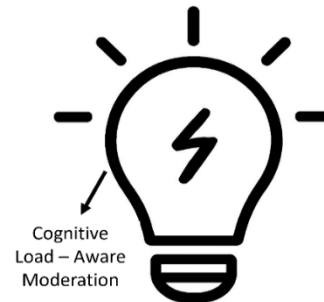


Figure 16: cognitive load aware

4.15 Satellite-aided threat localisation

This technique integrates geospatial intelligence into a threat detection system for global platforms. By using satellite data regional mapping, it helps localise malware outbreaks, misinformation surges and coordinated abuse campaigns. The system supports geo-fenced moderation, tailoring responses based on location. It is especially effective during disasters and regional crises, where misinformation can spread rapidly. This approach enhances situational awareness while maintaining privacy and scalability across diverse geographic zones Figure 17.

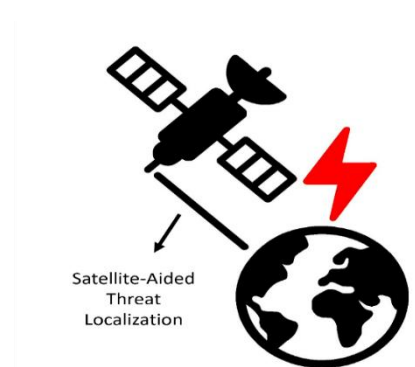


Figure 17: Satellite-aided threat localisation

4.16 Composable AI pipelines

The below Figure 18 composable AI pipelines allow developers to build a modular, plug-and-play threat detection system, instead of relying on a fixed architecture. Components such as CNNs, transformers. The federated nodes can be swapped in and out depending on the task, whether it is image-based malware detection, language toxicity analysis or privacy-preserving inference. The Flexibility of tailoring detection strategies to specific domains and data types. The scalability indicates easily expand and refine pipelines as threats evolve. The customisation optimises performance by selecting the most effective model for each context.

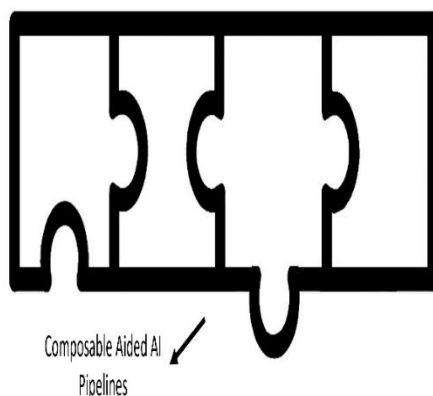


Figure 18: Composable AI pipelines

4.17 Emotion-aware toxicity filters

The traditional moderation system often relies on keyword matching and rule-based filters to detect toxic content. However, this approach misses subtle emotional cues such as sarcasm, passive aggression and manipulative tone, of common in cyberbullying and psychological abuse. The emotion-aware filter uses affective computing. A field that combines machine learning with emotional intelligence to analyse. Tone and sentiment in the text of a hostile sarcasm, veiled insults. The contextual

cues across conversations of repeated passive-aggressive replies. The emotion embeddings from models trained on annotated datasets with emotional labels.

This technique integrates geospatial intelligence into a threat detection system for global platforms. By using satellite data regional mapping, it helps localise malware outbreaks, misinformation surges and coordinated abuse campaigns. The system supports geo-fenced moderation, tailoring responses based on location. It The especially effective during disasters and regional crises, where misinformation can spread rapidly. This approach enhances situational awareness while maintaining can spread rapidly. This approach enhances situational awareness while maintaining privacy and scalability across diverse geographic zones Figure 19.

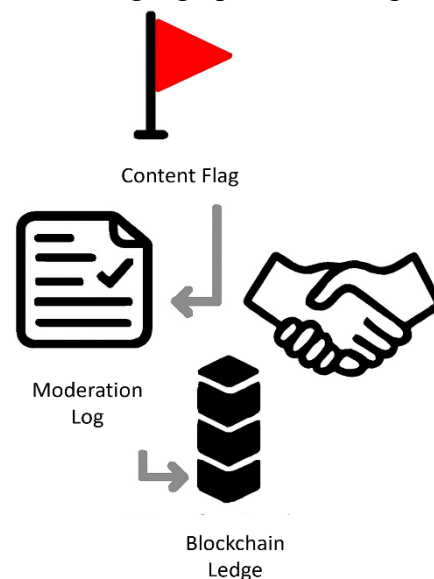


Figure 19: Privacy layout for next generation

As quantum computing advances and traditional encryption methods such as RSA and ECC may become vulnerable to quantum attacks. Quantum-resistant privacy layers proactively address this risk by integrating post-quantum cryptography (PQC) into systems that handle sensitive data. Federated learning allows models to train across

decentralised data sources without exposing raw data. The quantum encryption ensures that even if intercepted, shared model updates and metadata remain secure against future quantum decryption. The benefits of long-term privacy resilience across global platforms. The future proofing for compliance and secure collaboration Figure 19. The safe cross-border data to exchange in high-risk environments.

4.18 Ethical drift monitoring

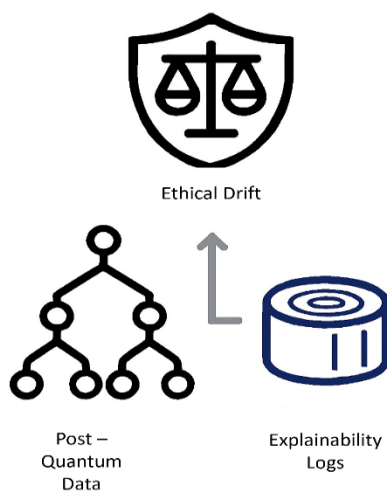


Figure 20: Enterprise AI architecture

Figure 20 indicates an ethical drift monitoring for enterprise AI. This illustrates how an AI system tracks ethical consistency over time using an audit mechanism. The ethical drift shield represents the system's commitments to fairness and accountability. The use of symbolic scales to indicate ethical balance. The decision tree diagram visualises how model decisions evolve across versions. The branching paths reflect changes in decision boundaries and logic. The captures interpretable output of SHAP values and attention maps for each model version. It enables across time and demographic groups. The indicate of insights from drift detection feedback into retraining and governance

modules. The support continues to improve and ensure regulatory compliance.

The track model decisions evolve over time. The use of versioned explainability logs to detect ethical drift and unintended bias. The supports regulatory audits and long-term accountability. The ethical drift occurs when an AI model's decisions gradually shift away from its original ethical standards, often due to retraining, data drift and evolving deployment contexts. The ethical drift monitoring helps detect and correct these shifts before they cause harm and violate compliance. The versioned explainable logs indicate a store model's decisions alongside interpretable explanations of SHAP, LIMD, and attention maps across model versions and time periods. The bias and fairness audits compare outputs across demographic groups and sensitive attributes to detect unintended bias and discriminatory patterns. The temporal drift analysis tracks how decision boundaries and feature importance change over time, flagging ethical inconsistencies. The regulatory support enables transparent reporting for audits, certifications and long-term accountability in high-stakes domains such as finance, healthcare and law enforcement. The main benefits of early detection of ethical violations and model misalignment. The support governance framework and AI accountability. The building trust with the users, regulators and stakeholders.

4.19 Blockchain-backed moderation logs

To ensure tamper-proof moderation and store flagged content and moderation actions on a distributed ledger. The builds trust with users and regulators through transparent and immutable records. This approach uses blockchain technology to create tamper-proof

records of moderation actions. Every post, user report and moderator decision is stored on a distributed ledger. The immutable indicates that once an recorded an entries cannot be altered or deleted. The transparency indicates an accessible for audits by using regulators and oversight bodies. The trustworthy builds confidence in platform fairness and accountability.

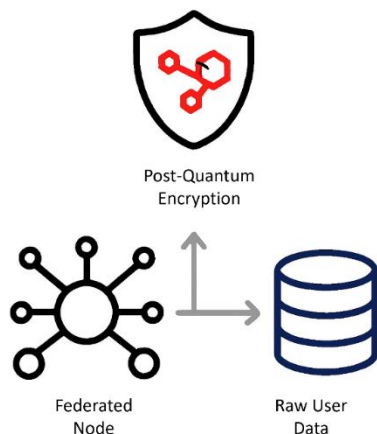


Figure 21: Blockchain-backed moderation

4.20 Advanced technical modules

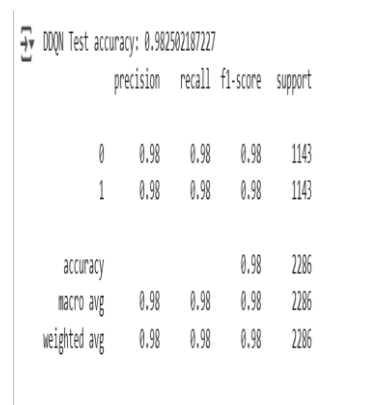
The below Table 1 represents an analysis of text content to flag abusive language, veiled insults, and manipulative phrasing. The transformers excel at contextual understanding, CNNs offer fast pattern recognition, and sarcasm detectors help catch emotionally deceptive toxicity. The process images to detect harmful content like graphic violence, hate symbols and doctored media. The CNNs are efficient for local features if vision transformers handle global context, and adversarial shields defend against evasion tactics. Goes beyond keyword detection to assess emotional tone, sarcasm and psychological manipulation. The affective embeddings encode emotional cues while sentiment models classify mood and intent. Building a social graph to detect coordinated

campaigns, botnets and influence networks. The graph neural network model relationship, while considering emotional weight edges and the psychological context of interactions. The uses of geospatial intelligence to detect region-specific threats like misinformation surges and disaster-related abuse. The satellite feeds provide real-time context, and overlays help apply geo-fenced moderation policies. Enables privacy-preserving learning across decentralised data sources. The post-quantum cryptography secures model updates, and explainability logs ensure transparency for audits and ethical drift monitoring.

Modules	Functions	Components
Text toxicity filter	Detect hate speech	Transformer
Image threat detection	Misinformation	CNN
Emotion-aware filter	Passive aggression	Sentiment models
Graph intelligence engine	Maps coordinated	Emotion weight edges
Federated node	Model training	PQC encryption

Table 1: Feature extraction module

5. Evaluation metrics



	precision	recall	f1-score	support
0	0.98	0.98	0.98	1143
1	0.98	0.98	0.98	1143
accuracy			0.98	2286
macro avg	0.98	0.98	0.98	2286
weighted avg	0.98	0.98	0.98	2286

Figure 22: Performance metrics of the DDQN model on test data

The above Figure 22 represents an explanation of the result in a table

performance evaluation of the DDQN model test. The test accuracy of the approximate value of 0.9825 and 98.25% of the model predicts malware in benign samples with very high accuracy. The precision value of 0.98 out of all the samples predicted as malware, using a benign rate of 98% was the correct value. The recall values of 0.98 out of all the actual malware of the benign sample of 98% were correctly detected. The F1 score value of 0.98 is balanced between precision and recall, which shows a strong reliability. The support values of each class of 0 is equal to benign, and 1 is equal to malware. The 1143 values of sample data for testing with the total values of 2286. The macro average and weighted average of both values as 0.98 to confirm the performance is consistent across both classes without the bias value. The DDQN model's state-of-the-art performance of 98% across all metrics is making a highly effective malware detection with balanced data handling.

Source	Types of features	Methods
Malware	API calls	Dynamic and static analysis
Malware	Opcodes	Disassembly tools
Malware	Network traffic	Wireshark
Malware	Behavioral logs	Sandbox execution
Toxic comments	Raw text	Text crawling

Table 2: Source of Malwares

The **Table 2** represents that the source of malware indicates the features as API calls, Opcodes, Network traffic, and behavioural logs. The extraction method such as static analysis and dynamic analysis. The Disassembly tools and packet capture of Wireshark are used in the extraction method. The behavioural logs of a sandbox execution

method. The toxic comments of a raw text in the data import and text crawling.

The following Figure 23 shows a confusion matrix performance of the DDQN malware detection model can be represented. The True value of 0 indicates the benign sample of 1123 was correctly classified, with 20 misclassified as malware. The True value of 1 in a malware sample of 1123 was correctly classified as 20 misclassified as benign. The overall performance has of very low misclassification of only 20 errors in each class out of 1143 samples. The high reliability and balanced detection with accuracy close to 98% for both malware and benign classes.

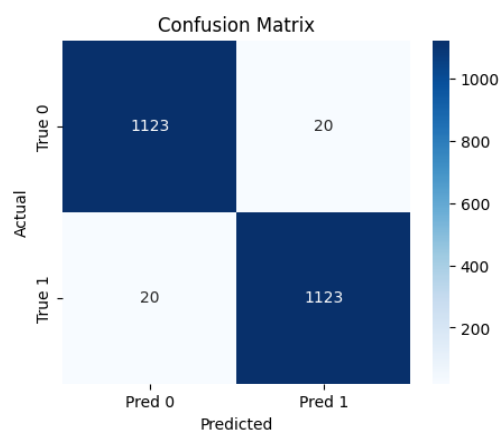


Figure 23: Confusion matrix of the DDQN model for malware detection

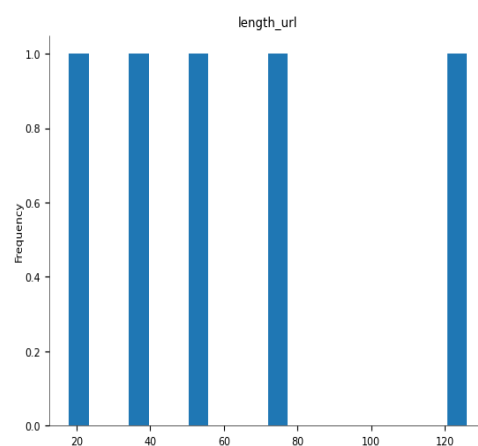


Figure 24: Distribution of URL length in data

The above Figure 24 represents a bar chart indicating of frequency distribution of URL length across the data. The X-axis shows different ranges of URL length of 20,40,60,80,120 characters. The Y-axis shows an indication of the frequency of how often the URL of a certain length appears. The chart reveals that the URLs are distributed across a variety of lengths, with a noticeable frequency around short values of 20 to 40 and longer values of 80 to 120 characters of length. The URL length is an important feature in malware and phishing detection, as an unusually long or short URL often indicates a suspicious or malicious action.

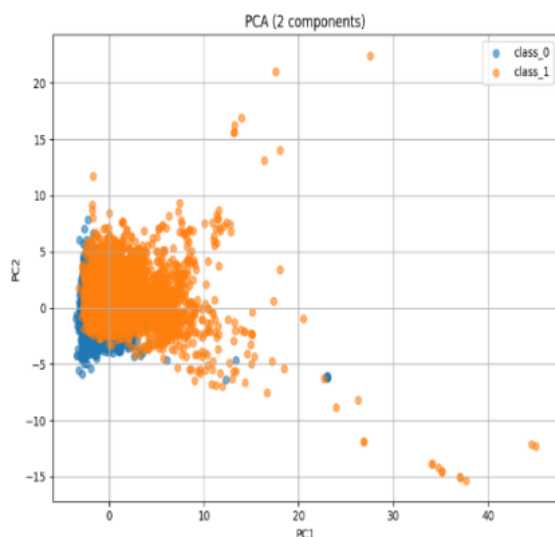


Figure 25: PCA-Based visualisation of the class using components

The above Figure 25 represents a PCA-based visualisation of class separation using two principal components. The figure shows a 2D scatter plot of high-dimensional data reduced using principal component analysis stands for PCA. The two principal components of PC1 and PC2 capture the most significant variables in the dataset. The class 0 of blue and the class 1 of orange are plotted to visualise separability. The clustering near the original suggests a shared structure while

spreading along PC! Indicates a variable between classes. The visualisation helps assess how well the feature distinguishes between classes and whether further dimensionality reduction of feature engineering is needed.

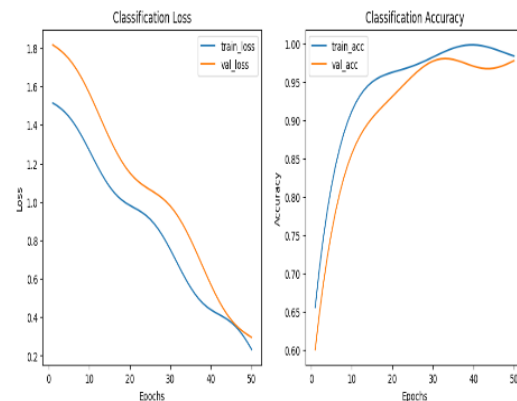


Figure 26: Training progress of the classification model

The above Figure 26 represents training progress data of the classification model to evaluate the left panel of classification loss. It shows a training loss of blue colour, and the validation loss is indicated as an orange colour. The decreasing value over 50 epochs. The indicated that the model is learning efficiently and generated as the well performanmced with no signs of overfitting. The right panel of the classification accuracy of the training data as blue and the validation accuracy as orange. To improve the upward trends of the model for more accurate prediction of both training and unseen validation data.

6. Experimental results

The below Figure 27 indicates that the user-friendly interface of the URL malware detection system with two main components. The left panel of the input data section of the user can enter the website address into the Enter URL field and click the Check here button to initiate a safety check. The right output after analysing the system display of

the safety score. The website is 88% safe to use and provides continuous data on button user action based on the result. The interface supports real-time data threat detection that helps the website. The practical example of integrating cybersecurity tools into an accessible digital platform.



Figure 27: User interface for URL malware detection with safety output

The following Figure 28 represents a user-facing interaction of the design for real-time URL malware detection. The two main components of the left panel input section of the user enter website address into the enter URL field and initiate a scan by clicking the check here button. The gradient styling enhanced visibility and user engagement. The right panel output after analysing the system displays a safety score of website is 0% safe to use, along with a continue button. The feedback helps the user make an informed decision about whether to proceed.

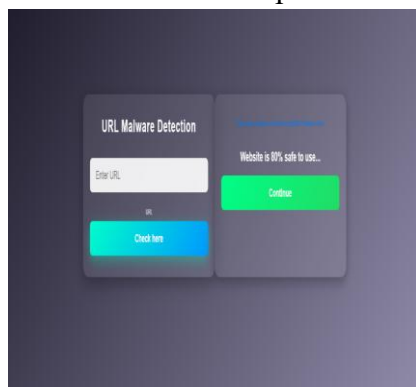


Figure 28: Interactive interface for URL safety detection

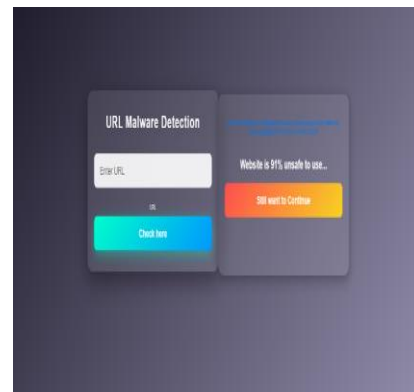


Figure 29: URL malware detection interface with time safety feedback

The above figure shows that Figure 29 represents the URL malware detection interface with real-time safety feedback. The web-based interface for assessing the safety URL that shows a left panel input section where of user can enter a website address into the Enter URL field and initiate a scan using the check here button. The Right panel output section of the system display a safety score of website is 51% to use. It provides the continue button to guide the user's action.

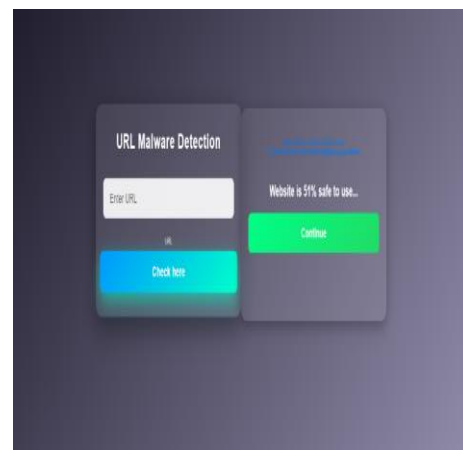


Figure 30: URL malware detection interface with high-risk warning and user override option

The above Figure 30 represents a URL malware detection interface with the high-risk warning user override option. The

cybersecurity interface is designed to assess the safety of a URL in real time. The left panel input section of the user to enter a website address and initiate a scan using the check here button with a cyan to green gradient for visual emphasis. The right output section of the system flags the URL of 91% of unsafe. The display of a red warning message and offering a still want to continue button. Allows the user to override the system recommendation if necessary. The interface highlights the importance of user awareness and control in the threat detection system of balancing automated risk assessment with informed decision-making.

7. Conclusion

The hybrid framework study of Double Deep Q-Networks of DDQN for malware detection and Bi-LSTM for toxic comment analysis was proposed. The methodology combined reinforcement learning for adaptive threat recognition with deep sequential modeling for natural language processing. To create a unified capability of addressing both technical threats of malware and social threats of toxic content. The result of the DDQN model achieved an accuracy of 98.25% with high precision, recall, and F1 score to prove its effectiveness in detecting polymorphic and zero-day malware. The Bi-LSTM model effectively captured subtle toxic patterns in textual data. The system offered a robust and reliable decision-making module that threads into four categories of malware, such as toxic only, combined threats, and safe communication. The result confirms that the proposed approaches of only enhance cybersecurity resiliences but also address the growing issues of online toxicity. The offering of multi-layered defenses of a mechanism in future work can focus on extending this hybrid

to additional domains, such as phishing detection, spam filtering, and adversarial attack on the real-time performance of large-scale deployment of the ecosystem.

8. REFERENCE

1. M. Wazid, A. K. Das, J. J. P. C. Rodrigues, S. Shetty and Y. Park, "IoT Malware Detection Approaches: Analysis and Research Challenges," in *IEEE Access*, vol. 7, pp. 182459-182476, 2019, doi: 10.1109/ACCESS.2019.2960412.
2. Lin, SY., Chien, SY., Chen, YZ. et al. Combating Online Malicious Behavior: Integrating Machine Learning and Deep Learning Methods for Harmful News and Toxic Comments. *Inf Syst Front* (2024). <https://doi.org/10.1007/s10796-024-10540-8>
3. Kattamuri, Santosh Jhansi, Ravi Kiran Varma Penmatsa, Sujata Chakravarty, and Venkata Sai Pavan Madabathula. "Swarm optimization and machine learning applied to PE malware detection towards cyber threat intelligence." *Electronics* 12, no. 2 (2023): 342. <https://doi.org/10.3390/electronics12020342>
4. Amira, Abdelouahab, Abdelouahid Derhab, Elmouatez Billah Karbab, and Omar Nouali. "A survey of malware analysis using community detection algorithms." *ACM Computing Surveys* 56, no. 2 (2023): 1-29. <https://doi.org/10.1145/3610223>
5. K. Aryal, M. Gupta, M. Abdelsalam, P. Kunwar, and B. Thuraisingham, "A Survey on Adversarial Attacks for Malware Analysis," in *IEEE Access*, vol. 13, pp. 428-459, 2025, doi: 10.1109/ACCESS.2024.3519524.
6. N. N. Kumar et al., "Stacked Bi-LSTM for Advanced Toxicity Detection in Comment Classification," 2024 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAET), Kota Kinabalu, Malaysia, 2024, pp. 337-342, doi: 10.1109/IICAET62352.2024.10729970.
7. Ahuja, R., Banga, A., Sharma, S.C. (2021). Detecting Abusive Comments Using Ensemble Deep Learning Algorithms. In: Stamp, M., Alazab, M., Shalaginov, A. (eds) *Malware Analysis Using*

- Artificial Intelligence and Deep Learning. Springer, Cham. https://doi.org/10.1007/978-3-030-62582-5_20
8. M. Amin et al., "Security and privacy awareness of smartphone users in Indonesia," J. Phys. Conf. Ser., vol. 1882, no. 1, p. 12134, May 2021, doi: 10.1088/1742-6596/1882/1/012134.
9. Gandhi, H., Bachwani, R., Nanade, A. (2023). Detecting Toxic Comments Using FastText, CNN, and LSTM Models. In: Singh, M., Tyagi, V., Gupta, P., Flusser, J., Ören, T. (eds) Advances in Computing and Data Sciences. ICACDS 2023. Communications in Computer and Information Science, vol 1848. Springer, Cham. https://doi.org/10.1007/978-3-031-37940-6_20
10. Marshan, A., Nizar, F.N.M., Ioannou, A. et al. Comparing Machine Learning and Deep Learning Techniques for Text Analytics: Detecting the Severity of Hate Comments Online. Inf Syst Front 27, 487–505 (2025). <https://doi.org/10.1007/s10796-023-10446-x>
11. Naseeba, B., Sai, P.H.R., Karthik, B.V.P., Chitteti, C., Sai, K., Avanija, J. (2023). Toxic Comment Classification. In: Abraham, A., Hong, TP., Kotecha, K., Ma, K., Manghirmalani Mishra, P., Gandhi, N. (eds) Hybrid Intelligent Systems. HIS 2022. Lecture Notes in Networks and Systems, vol 647. Springer, Cham. https://doi.org/10.1007/978-3-031-27409-1_80
12. Lin, SY., Chien, SY., Chen, YZ. et al. Combating Online Malicious Behavior: Integrating Machine Learning and Deep Learning Methods for Harmful News and Toxic Comments. Inf Syst Front (2024). <https://doi.org/10.1007/s10796-024-10540-8>
13. Akhtar, M. S., & Feng, T. (2022). Malware Analysis and Detection Using Machine Learning Algorithms. Symmetry, 14(11), 2304. <https://doi.org/10.3390/sym14112304>
14. Paoletti GGioacchini LMellia MVassio LAlmeida J(2025)CoDÆN: Benchmarks and Comparison of Evolutionary Community Detection Algorithms for Dynamic Networks Transactions on the Web 10.1145/371898819:3(1-25)Online publication date: 22-Aug-2025 <https://dl.acm.org/doi/10.1145/3718988>
15. Risch, J., Krestel, R. (2020). Toxic Comment Detection in Online Discussions. In: Agarwal, B., Nayak, R., Mittal, N., Patnaik, S. (eds) Deep Learning-Based Approaches for Sentiment Analysis. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-15-1216-2_4