Algorithms for Intelligent Systems
Series Editors: Jagdish Chand Bansal · Kusum Deep · Atulya K. Nagar

Vishal Goar
Aditi Sharma
Jungpil Shin
M. Firoz Mridha *Editors*

Deep Learning and Visual Artificial Intelligence

Proceedings of ICDLAI 2024

Springer

Algorithms for Intelligent Systems

This book series publishes research on the analysis and development of algorithms for intelligent systems with their applications to various real world problems. It covers research related to autonomous agents, multi-agent systems, behavioral modeling, reinforcement learning, game theory, mechanism design, machine learning, meta-heuristic search, optimization, planning and scheduling, artificial neural networks, evolutionary computation, swarm intelligence and other algorithms for intelligent systems.

The book series includes recent advancements, modification and applications of the artificial neural networks, evolutionary computation, swarm intelligence, artificial immune systems, fuzzy system, autonomous and multi agent systems, machine learning and other intelligent systems related areas. The material will be beneficial for the graduate students, post-graduate students as well as the researchers who want a broader view of advances in algorithms for intelligent systems. The contents will also be useful to the researchers from other fields who have no knowledge of the power of intelligent systems, e.g. the researchers in the field of bioinformatics, biochemists, mechanical and chemical engineers, economists, musicians and medical practitioners.

The series publishes monographs, edited volumes, advanced textbooks and selected proceedings.

Indexed by zbMATH.

All books published in the series are submitted for consideration in Web of Science.

Vishal Goar · Aditi Sharma · Jungpil Shin ·
M. Firoz Mridha
Editors

Deep Learning and Visual Artificial Intelligence

Proceedings of ICDLAI 2024

Springer

Editors

Vishal Goar
Engineering College Bikaner
Bikaner Technical University
Bikaner, Rajasthan, India

Jungpil Shin
Pattern Processing Lab, School
of Computer Science and Engineering
University of Aizu
Aizu-Wakamatsu, Fukushima, Japan

Aditi Sharma
Department of Computer Science
and Engineering, Symbiosis Institute
of Technology
Symbiosis International University
Pune, Maharashtra, India

M. Firoz Mridha
Department of Computer Science
American International University
Dhaka, Bangladesh

ISSN 2524-7565 ISSN 2524-7573 (electronic)
Algorithms for Intelligent Systems
ISBN 978-981-97-4532-6 ISBN 978-981-97-4533-3 (eBook)
https://doi.org/10.1007/978-981-97-4533-3

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

The conference proceedings volume for the 1st International Conference on Deep Learning and Visual Artificial Intelligence (ICDLAI-2024) held at Bikaner Technical University in Rajasthan, India, on March 16–17, 2024, contains the written versions of research contributions that were accepted and presented during the event. The primary objective of ICDLAI-2024 was to provide a scholarly platform for academicians, engineers, and researchers to showcase their cutting-edge research and innovative work within the expansive domains of deep learning and artificial intelligence.

Throughout the conference, a diverse range of interactive forums, core technical sessions, and top-tier research articles were presented, highlighting the ongoing efforts within deep learning and artificial intelligence to develop new ideas, methodologies, and tools to address pertinent issues. The conference proceedings encompass a wide array of topics, including system paradigms, approaches, and technical reviews that leverage knowledge and intelligence across various domains.

ICDLAI-2024 received over 257 submissions from 15 different nations, including the United States, Iraq, China, Ghana, the United Kingdom, Bangladesh, Kazakhstan, and more. Each submission underwent a rigorous plagiarism check and was reviewed by at least three reviewers, with some entries undergoing additional reviews. Ultimately, 40 outstanding articles were selected for publication in this proceedings volume, representing an acceptance rate of 15.56%.

We extend our sincere thanks to all participants for their invaluable contributions to the conference program and the proceedings. Additionally, we express our gratitude to the reviewers for their insightful feedback on the papers and acknowledge the hard work of the organizing team members.

Bikaner, India — Vishal Goar
Pune, India — Aditi Sharma
Aizu-Wakamatsu, Japan — Jungpil Shin
Dhaka, Bangladesh — M. Firoz Mridha

Contents

About the Editors

Dr. Vishal Goar is currently working as an Assistant Professor at Engineering College Bikaner, he is the former Dean of Research at Bikaner Technical University, Bikaner. He has wide experience of 17 years as an academician, a researcher, and in industry. He has published 46 research papers in international journals (SCI/ Scopus/WoS) and presented 41 papers at national/international conferences. He has edited/authored 11 books including reference, text, and edited books of Springer, Singapore, ACM USA, and Scholars-Press, Germany. He has published two patents and 3 patent grants to his credit. He supervised 6 Ph.D. research scholars in areas of cloud computing, software development, neural networks, and deep learning. He also served as an organizing chair, the convener, and the publicity chair of various Springer/IEEE/ACM internationally renowned conferences such as AICTC and ICACCA. His research areas include Artificial Intelligence, Cloud Computing, IoT, and Deep Learning. He is a life member of Professional Bodies such as IEANG, ISTE, IACSIT, UACEE, CSTA, SDIWC, and CSI. He has served as a technical program committee (TPC) member and a reviewer in various Springer, IEEE international conferences, ICACE, ICCECT, ICCSIE, and many more. He was invited as an invited speaker/resource person in TEQIP III Faculty Development Programs and delivered keynotes at international conferences such as SmartCom-2020. He is also serving as a member of the Board of Studies for prestigious universities such as Bikaner Technical University, Bikaner, Maharaja Ganga Singh University, Bikaner, and Tantia University, Sriganganagar. He is a member of the Research Board at Bikaner Technical University, Bikaner. He is also working as an editorial board member, an advisory board member, and a reviewer of many international journals.

Dr. Aditi Sharma holds a degree of B.Tech. from Mody Institute of Technology and Science, Lakshmangarh, and a Ph.D. degree from MBM Engineering College, JNVU, India. She is working as an associate professor at Symbiosis Institute of Technology, Symbiosis International University, Pune. She has 15 years of teaching experience and also worked as a post-doctoral fellow at the School of Engineering and Digital Sciences at Nazarbayev University, Kazakhstan, in the area of Intelligent Cryptosystems, IoT, and cloud in Robotics. She is the visiting faculty at Astana

IT University, Kazakhstan, and the University of Uyo Nigeria. Dr. Aditi Sharma has published 65 research papers in international journals SCI, ESCI, Scopus, and national/international conferences and authored four books and 19 patents. She is an active member of the IEEE 5G security group and is the IEEE senior member and a life member of Professional Bodies such as the Cryptology Society of India, IEANG, and N2women Society. Her research area includes cryptography, blockchaining, VLSI, cellular automata, machine learning, AI chatbots, IoT, and artificial intelligence. Dr. Aditi Sharma organized various workshops, industrial visits, and conferences during her career. She achieved many awards and scholarships including the Nav Shakti Award from the North Eastern Council, Ministry of DoNER, and the Government of India; Parvati-Chanda Devi and Gargi scholarships and Awards from Madan Lal Khurana, Sunil Dutt, and A. P. J. Abdul Kalam. She also got the Best Teacher Award by the Rajasthan Patrika Group.

Jungpil Shin (Senior Member, IEEE) received a B.Sc. in Computer Science and Statistics and an M.Sc. in Computer Science from Pusan National University, Korea, in 1990 and 1994, respectively. He received his Ph.D. in computer science and communication engineering from Kyushu University, Japan, in 1999, under a scholarship from the Japanese government (MEXT). He was an Associate Professor, a Senior Associate Professor, and a Full Professor at the School of Computer Science and Engineering, The University of Aizu, Japan in 1999, 2004, and 2019, respectively. His research interests include pattern recognition, image processing, computer vision, machine learning, human-computer interaction, nontouch interfaces, human gesture recognition, automatic control, Parkinson's disease diagnosis, ADHD diagnosis, user authentication, machine intelligence, bioinformatics, as well as handwriting analysis, recognition, and synthesis. He is a member of ACM, IEICE, IPSJ, KISS, and KIPS. He served as general chair, program chair, and committee for numerous international conferences. He serves as an Editor of IEEE journals, Springer, Sage, Taylor & Francis, MDPI Sensors and Electronics, and Tech Science. He serves as an Editorial Board Member of Scientific Reports. He serves as a reviewer for several major IEEE and SCI journals. He has co-authored more than 400 published papers for widely cited journals and conferences.

Dr. M. Firoz Mridha (the senior member, IEEE) is currently working as an associate professor in the Department of Computer Science at American International University-Bangladesh (AIUB). Before that he worked as an associate professor and the chairman in the department of CSE of Bangladesh University of Business and Technology. He also worked as a CSE department faculty member at the University of Asia Pacific and as a graduate head from 2012 to 2019. He received his Ph.D. in AI/ML from Jahangirnagar University in the year 2017. His research experience, within both academia and industry, results in over 120 journal and conference publications. His research work contributed to the reputed Journal of Scientific Reports–Nature, Knowledge-Based Systems, Artificial Intelligence Review, IEEE Access, Sensors, Cancers and Applied Sciences, etc. His research interests include artificial intelligence (AI), machine learning, deep learning, and natural language processing

(NLP). For more than 10 (Ten) years, he has been with the master's and undergraduate students as a supervisor of their thesis work. His research interests include artificial intelligence (AI), machine learning, natural language processing (NLP), big data analysis, etc. He has served as a program committee member in several international conferences/workshops. He served as an associate editor of several journals including *PLOS ONE* Journal. He has served as a reviewer of reputed journals and many conferences.

Chapter 1
Safeguarding User Privacy: Machine Learning Strategies for Android Malware Detection

R. Thamizharasi and K. Chitra

1 Introduction

Google Android's imperious presence in the smartphone industry, with uplift of 75% market share and a user community of 2.8 billion, emphasizes its attractiveness to both users and manufacturers. This is assigned to its open-source design and flexibility, making it a preferred choice for creating cost-effective smart devices. The Android SDK's accessibility has attracted developers due to its efficiency in creating applications for this extensive user base. Nevertheless, the platform's pervasive popularity has also rendered to malicious activities, notably the accumulation of malware apps. Over the past decades, the number of malware instances has surged significantly, reaching 1.34 billion as reported by AV-Test [1]. These malicious applications pose a significant threat by targeting user privacy and sensitive data. Thus, developing a productive method to detect and combat these malicious applications is of paramount importance. Machine learning arrived as a promising approach to tackle this challenge. A subset of artificial intelligence enables computer programs to automatically learn from data and make predictions. Through analyzing different features, such as API calls, requested permissions, and network activity, a machine learning model can diagnose patterns linked with malware behavior on Android devices. This technique, known as Android malware detection, enables the creation of models capable of differentiate between legitimate and malicious applications.

R. Thamizharasi (✉)
Research Scholar, Department of Computer Science, RVS College of Arts and Science, Coimbatore, Tamil Nadu, India
e-mail: vprtamil@gmail.com

K. Chitra
Assistant professor, Department of Computer Science, RVS College of Arts and Science, Coimbatore, Tamil Nadu, India

V. Goar et al. (eds.), *Deep Learning and Visual Artificial Intelligence*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-97-4533-3_1

An essential aspect of this strategy includes monitoring the network to which an Android device is connected. This helps to assess the suspiciousness of applications. Following the trained machine learning model, it can be used to classify new applications as either malicious or non-malicious automatically. Keyes et al. conducted a study in 2021 and discovered that engrossing the random forest and ensemble techniques yielded the highest accuracy in detecting malware [2]. Twofold cross-validation was used to assess the model's performance, accounting for various measures like F1-score, precision, and recall. By hitching up the capabilities of machine learning in Android malware detection, we can improve users' protection from the threats posed by malicious application to ensure their privacy and data security.

2 Survey Study

Within this section, we carried out a comprehensive survey related to Android malware detection prevailing the existing research and studies. The survey focused to provide a contextual understanding of the current landscape, highlighting the challenges, methodologies, and progressions in the field. By examining the existing literature, we gain insights into the state-of-the-art techniques and pinpoint gaps that our proposed approach intent to address.

2.1 Android Malware Detection Landscape

The widespread presence of Android as the foremost smartphone operating system has made it vulnerable to malware attacks [3].

2.2 Ensemble Techniques for Android Malware Detection

Ensemble techniques, which integrate multiple base classifiers to boost prediction accuracy, have assurance in Android malware detection [4]. Research has highlighted the efficacy of ensemble methods like random forest in accomplishing high accuracy rates. The diversity introduced by ensemble methods conduces to improved generalization and robustness.

2.3 Hyperparameter Optimization and Cross-Validation

Enhancing the enactment of machine learning models, optimizing hyperparameters plays a vital role [5]. Cross-validation techniques, like *k*-fold and twofold cross-validation, play an essential role in selecting optimal hyperparameters by assessing model performance on various subsets of the dataset [6]. These techniques avert overfitting and ensure the model's capacity to simplify to novel data.

2.4 Research Gaps and Challenges

While prior research has made significant contributions to Android malware detection, there are still challenges that need to be addressed [7]. Some of these challenges encompass the rapid evolution of malware techniques, the imperative for real-time detection, and the balance between accuracy and computational efficiency. Recognizing these gaps provides the direction for our proposed methodology.

3 Proposed Methodology—Machine Learning Strategies for Android Malware Detection

Figure 1 shows the steps involved in proposed research methodology.

3.1 Data Collection

During the stage of data collection as explored in Fig. 2, a diverse dataset of Android malware samples is combined, depicting different types like adware, scareware, and SMS malware. The dataset is enriched with legitimate Android applications for comparative analysis. Thorough care is dedicated to choosing data sources to ensure

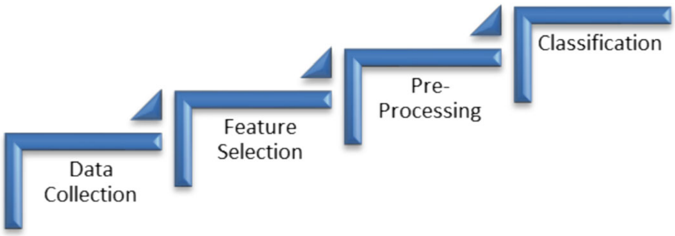
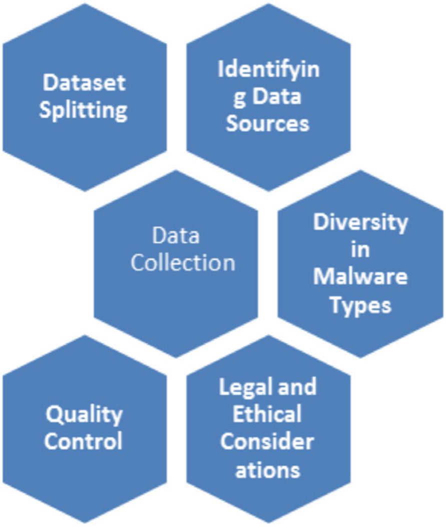


Fig. 1 Block diagram of machine learning strategies for Android malware detection

Fig. 2 Data collection



quality and legality. Data preprocessing may be entailed to clean and format the data as needed; if necessary, data balancing techniques are applied to address any class imbalance. The resulting dataset forms the basis for training and evaluating machine learning models for detection of Android malware, ensuring their accuracy and applicability to real-world scenarios.

3.2 Feature Selection

In the context of Android malware detection, feature selection is relevant to construct an efficient and accurate model which is able to discerning between malicious and non-malicious applications. To accomplish this, two statistical tests are employed for different feature types. Initially, the Mann–Whitney *U* test is utilized for numerical features. This nonparametric test assesses whether two independent samples have distinct distributions. In the context of Android malware detection, the numerical features might represent numerous characteristics of the applications, such as the number of API calls, permission requests, or other behavioral patterns. Employing the Mann–Whitney *U* test to these numerical features, we can determine their significance in differentiating between malware and legitimate apps. Features that performing significant differences in their distributions between the two classes are deemed more relevant and informative for the classification process [8, 9].

The Mann–Whitney *U* test is given to compare two independent samples and assess whether they originate from the same distribution. In the context of Android malware detection, let us consider a numerical feature "*X*" that illustrates the characteristic of the applications, such as the number of API calls made by an app. We have two sets of samples: "*M*" for malware applications and "*N*" for non-malicious (legitimate) applications. The null hypothesis ($H0$) assumes that the distributions of "*X*" in both sets are the identical, while the alternative hypothesis ($H1$) assumes that they differ.

The Mann–Whitney *U* statistic "*U*" is calculated as follows:

$$U = \min(U1, U2) \tag{1}$$

Here, $U1$ is the sum of ranks of the samples in set "*M*," and $U2$ is the sum of ranks of the samples in set "*N*."

The *p*-value connected with the Mann–Whitney *U* test is used to drive the significance of the difference between the two distributions. If the *p*-value is under predefined significance level (e.g., 0.05), we reject the null hypothesis and conclude that the feature "*X*" is significant in differentiating between malware and non-malicious applications. In this case, "*X*" is deemed an informative feature for the classification process.

Secondly, the chi2 test is used for categorical features. The chi2 test is a statistical method that examines the independence between categorical variables. In the context of Android malware detection, categorical features could encompass attributes such as the presence or absence of specific permissions, the app's category, or other discrete properties. By dominating these categorical features to the chi2 test, we can measure their significance in distinguishing between malware and non-malware applications. Features that represent strong dependencies or notable differences in distribution between the two classes are considered more valuable for the classification task.

Chi2 Test for Categorical Features:

The chi2 test is used to appraise the independence between categorical variables. Let us take a categorical feature "*Y*" that symbolized an attribute of the applications, such as the presence or absence of a specific permission requested by the app. We have two sets of samples: "*M*" for malware applications and "*N*" for non-malicious applications. A contingency table is created to show the frequency of each category in both sets as explored in Table 1.

The chi2 statistic "χ^2" is calculated as follows:

$$\chi^2 = \sum\big((Ni1 * Nj1 - Ni2 * Nj2)^2 / (Ni1 + Ni2) * (Nj1 + Nj2)\big) \tag{2}$$

Here, $Ni1$ and $Ni2$ are the frequencies of category "*i*" in sets "*M*" and "*N*," respectively, and $Nj1$ and $Nj2$ are the frequencies of category "*j*" in sets "*M*" and "*N*," respectively.

Table 1 Contingency table

Category-1	Category_1	Category_*k*
Malware (M)	$N11$	$N1k$
Non-malicious	$N21$	$N2k$

The *p*-value connected with the chi2 test is used to gage the significance of the dependency between the feature "*Y*" and the application's class (malware or non-malicious). If the *p*-value falls under the significance level, we reject the null hypothesis of independence, indicating that the feature "*Y*" is informative and exhibits a significant difference in distribution between the two classes.

By utilizing the Mann–Whitney *U* test for numerical features and the chi2 test for categorical features, the feature selection process effectively recognizes the most important attributes that plays a significant role in detection of Android malware. These selected features are pivotal for constructing a compact and robust model that can accurately differentiate between malicious and non-malicious applications, enhancing the overall performance and efficiency of the Android malware detection system.

3.3 Preprocessing

It is a necessary step in machine learning to transform raw data into a format suitable for model training. In this context, we will interpret the preprocessing steps, specifically one-hot encoding and scaling, with mathematical models. By applying one-hot encoding to categorical data and scaling numerical features, the data are preprocessed into a format that is tractable to machine learning algorithms, enabling the model to make precise predictions and classifications based on processed data [10].

3.4 Classification

Let us break down the steps enhanced in the improved machine learning-based classification for Android malware utilizing the random forest algorithm and ensemble-based classification methods (RF-EBCM), along with the optimization process using twofold cross-validation [11].

Random Forest Algorithm: This algorithm is used to create a decision tree based on given data. The division of labor is as follows:

$$\text{RandomForest }(X) = \text{mode}(\{T1(X), T2(X), \ldots, Tn\_\text{estimators}(X)\}) \quad (3)$$

Here, New models: $T1(X), T2(X), \ldots$ and $Tn_$estimators (X) represent the predictions from each decision tree in the random forest for sample "*X*." The most frequent class label in each decision tree prediction is created by the function type.

Ensemble-Based Classification Methods (RF-EBCM): This method requires combining classification methods (such as the random forest algorithm) into an ensemble based on base classifiers. The following represents the classification process:

$$\text{RF\_EBCM }(X) = \text{type}(\{C1(X), C2(X), \ldots, Cn\_\text{classifiers}(X)\}) \quad (4)$$

Here, X is "m," a new model with character. Predictions for instance "X" from all base classifiers in the set are represented by the letters $C1(X)$, $C2(X)$, … and $Cn\_$ classifiers (X). The type function provides a list of the most frequently occurring events in the class in the prediction of all base classifiers.

Optimization with Twofold Cross-Validation: In order to achieve the best performance in the ensemble, the random forest algorithm must be optimized by selecting the optimum set of hyperparameters. Twofold cross-validation is used to efficiently evaluate the model's performance with different hyperparameter combinations.

In twofold cross-validation, the dataset is divided into two equal-sized subsets (A and B). The optimization process works as follows.

In the initial iteration, subset A acts as the training set, and subset B acts as the testing set. The random forest hyperparameters are adjusted, and the model is trained on the training set. The performance metric (e.g., accuracy, $F1$-score) is calculated on the testing set.

In the second iteration, the process is reversed: Subset B acts as the training set, and subset A acts as the testing set. The random forest hyperparameters are again adjusted, and the model is trained on the training set. The performance metric is computed on the testing set.

The average performance metric across the two iterations is utilized to evaluate the random forest's performance with different hyperparameter combinations. The hyperparameters that result in the best average performance are chosen as the optimized set for the random forest.

Sample of Improved Machine Learning-Based Classification for Android Malware: Let us take an example where we have a dataset of Android applications categorized as either "Malware" or "Legitimate." The dataset contains diverse features, like the number of API calls, requested permissions, and network activity.

Random Forest Algorithm

We build a random forest with "n_estimators" = 100 (i.e., 100 decision trees). The random forest will learn from the training data and generate an ensemble of 100 decision trees, each capable of making predictions for novel Android applications.

Ensemble-Based Classification Methods (RF-EBCM)

We form an ensemble that includes the random forest as one of the base classifiers. The ensemble may also include other classifiers such as gradient boosting machines and support vector machines.

Optimization with Twofold Cross-Validation: We use twofold cross-validation to enhance the random forest's hyperparameters. The process involves splitting the dataset into two equal subsets denoted as A and B. We vary hyperparameters, such as "max_depth" and "min_samples_split," and train the random forest on subset *A*. We evaluate its performance on subset *B*. This process is then reversed roles involving training on *B* and testing on *A*.

After the twofold cross-validation, we evaluate the average performance of the random forest with various hyperparameter combinations. The optimal hyperparameters are selected to achieve optimal performance for the random forest.

By leveraging the optimized random forest and other base classifiers in the ensemble, the enhanced machine learning-based classification for Android malware can accurately identify and classify malicious applications, enhancing mobile security and safeguarding users from potential threats. The ensemble's diverse set of classifiers and the optimization process through twofold cross-validation ensure a resilient and effective model for Android malware detection. The steps involved in proposed research are explored in Table 2. Table 3 explicates the feature dataset.

This dataset includes four features:

- App Name: The name of the Android application.
- Number of API Calls: The number of API calls made by the application.
- Requested Permissions: The permissions requested by the application.
- Network Activity: The network activity of the application.

Table 2 The steps involved in the proposed research

Step	Description
1. Collect a dataset of Android applications labeled as either "Malware" or "Legitimate"	The dataset should contain various features, such as the number of API calls, requested permissions, and network activity
2. Build a random forest with "n_estimators" = 100 (i.e., 100 decision trees)	The random forest will learn from the training data and generate an ensemble of 100 decision trees, each capable of making predictions for new Android applications
3. Construct an ensemble that includes the random forest as one of the base classifiers	The ensemble may also include other classifiers like gradient boosting machines and support vector machines
4. Use twofold cross-validation to optimize the random forest's hyperparameters	The process involves dividing the dataset into two equal subsets (*A* and *B*). We vary hyperparameters, such as "max_depth" and "min_samples_split," and train the random forest on subset *A*. We evaluate its performance on subset *B*. We repeat the process with reversed roles (training on *B* and testing on *A*)
5. After the twofold cross-validation, we analyze the average performance of the random forest with different hyperparameter combinations	The best hyperparameters are selected to achieve optimal performance for the random forest

Table 3 Features of dataset

App name	Number of API calls	Requested permissions
Malware 1	100	SMS, Contacts, location
Malware 2	50	Internet, storage
Legitimate 1	20	None
Legitimate 2	30	Camera, microphone
Taking pictures and recording audio		Legitimate

The label column indicates whether the application is malware (Malware) or legitimate (Legitimate).

4 Result and Discussion

The model is evaluated by the performance metrics as accuracy, precision, recall, and F1-score as explored in Table 4.

Table 4 provides the performance of different classification models employing an ensemble-based classification method called RF-EBCM (random forest with ensemble-based classification method). Each row in the table represents a different combination of classifiers, preprocessors, and samplers. To evaluate the models, the metrics include $F2$-score, $F1$-score, recall, and precision. Here is a breakdown of the key elements.

RandomForestClassifier, XGBClassifier, and LGBMClassifier. Preprocessor: The method is used to preprocess the data before feeding it into the classifier. Common preprocessors encompass StandardScaler, MinMaxScaler, and RobustScaler. Sampler: The method is applied to balance the class distribution in

Table 4 Classification report using random forest with ensemble-based classification methods (RF-EBCM) with SMOTE sampler

	Classifier	Preprocessor	CV $F2$ Score	CV $F1$	CV recall	CV precision
0	RandomForestClassifier	StandardScaler	0.864462	0.874977	0.858342	0.895014
1	RandomForestClassifier	MinMaxScaler	0.851477	0.865639	0.843465	0.893071
2	RandomForestClassifier	RobustScaler	0.889085	0.891087	0.887785	0.894554
3	XGBClassifier	StandardScaler	0.868146	0.877535	0.862617	0.895288
4	XGBClassifier	MinMaxScaler	0.848002	0.863807	0.839386	0.895067
5	XGBClassifier	RobustScaler	0.864448	0.874268	0.858623	0.892721
6	LGBMClassifier	StandardScaler	0.871849	0.879234	0.867354	0.892845
	LGBMClassifier	MinMaxScaler	0.875484	0.881530	0.871739	0.892514
8	LGBMClassifier	RobustScaler	0.851393	0.865168	0.843505	0.891646

the dataset. In this case, the SMOTE technique is used, which stands for synthetic minority over-sampling technique. It creates synthetic samples of the minority class to balance the data classifier. The machine learning algorithm used for classification, such that.

CV $F2$-Score: The $F2$-score is a metric that amalgamates precision and recall. It emphasizes recall more than precision, which can be useful in cases where one class is more important to detect than the other.

CV $F1$, CV Recall, CV Precision: These are the traditional $F1$-score, recall, and precision metrics, respectively, calculated using cross-validation (CV) on the dataset.

The latter section of the table provides the classification report for the best-performing classifiers based on the CV $F2$-score. The models are ranked from top to bottom based on their CV $F2$-scores, indicating their overall effectiveness in distinguishing between classes (malicious and non-malicious applications). Each row shows performance metrics of the classifier using various preprocessors and samplers.

In conclusion, the table showcases the performance of different classifiers under various preprocessing and sampling techniques. These models can accurately classify applications as either malicious or non-malicious based on the provided features with the help of metrics. The goal is to identify the best combination of classifier, preprocessor, and sampler that yields the highest classification performance.

5 Conclusion

In summing up, improving Android malware detection through machine learning involves a general process. Different data collection methods were used to provide identical in a row about lawful and unlawful applications. The next step is specific assortment, which involves identify and collect data skin for accurate classification using statistical tests such as the chi2 test for categorical features and the Mann–Whitney U test for statistical differences. The data are ready for training through preprocessing methods such as feature scaling and single-bit coding. In the classification phase, prediction is made by combining the results of the classifier using the random forest classification and clustering method (RF-EBCM). Using twofold cross-validation, the optimization process tunes the random forest hyperparameters to achieve the best performance. All these methods together help create a powerful and effective model for detecting Android malware and improving digital security.

References

1. Wang X, Zhang L, Zhao K, Ding X, Yu M (2022) MFDroid: a stacking ensemble learning framework for Android malware detection. Sensors 22(7):2597

2. Keyes DS, Li B, Kaur G, Lashkari AH, Gagnon F, Massicotte F (2021)EntropLyzer: 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), Hamilton, ON, Canada, pp 1–12. https://doi.org/10.1109/RDAAPS48126.2021.9452002
3. Felt AP, Chin E, Hanna S, Song D, Wagner D (2011) Android permissions demystified. In: Proceedings of the 18th ACM conference on computer and communications security (CCS)
4. Rastogi V, Chen Y, Jiang X (2013) Catch me if you can: evaluating Android anti-malware against transformation attacks. In: Proceedings of the 2014 ACM conference on computer and communications security (CCS)
5. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Duchesnay É et al (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
6. Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13(Feb):281–305
7. Bilge L, Demir H, Dogan A, Peddabachagari S (2014) EXPOSURE: finding malicious domains using passive DNS analysis. In: Proceedings of the network and distributed system security symposium (NDSS)
8. Naway A, Li Y (2019) Android malware detection using autoencoder. arXiv preprint arXiv:1901.07315
9. Dhalaria M, Gandotra E (2024) MalDetect: a classifier fusion approach for detection of android malware. Expert Syst Appl 235:121155
10. Zhang X, Wang J, Xu J, Gu C (2023) Detection of Android Malware based on deep forest and feature enhancement. IEEE Access 11:29344–29359
11. Yao X, Li Y, Shi Z, Liu K, Du X (2023) Android malware detection based on sensitive features combination. Concurr Comput Pract Exp 35(6):1–1

Chapter 2
Deep Learning in Electronic Word-of-Mouth: A Comprehensive Review and Future Directions

Sneha Singh, Deepak Kaushal, Muskan Singh, and Sanjay Taneja

1 Introduction

As a result of the appearance of the Internet, in which individuals are not connected by physical interaction, a virtual society related via cyberspace has emerged. Individuals on the other end of the screen get a propensity for handshake with other people without in fact teaching them of confidence because of the absence of any bodily interactions between them [1]. The Internet made its entrance in the form of swapping data. The type of communication web 2.0 has fostered includes one-to-one, many-to-many, many-to-one, and one-to-many consumer discussions on the Internet regarding a product or company [2]. Therefore, through creating eWOM, many organizations have generated such customer-to-customer exchanges throughout the exchange and information experience of online product, program, brand, or organization information access to a large group of persons and companies, many prior nominees, or active clients [3]. In the age of the fourth industrial revolution, eWOM is a whirling advancement of WOM. Electronic word-of-mouth has quickly become a dominant, ubiquitous, and dynamic force in the consumer world. To be precise, online forums and personal blogs in the early days have given way to social platforms and unique idea-finding Web sites. According to [4], social networks are providing consumers a place of residence, permitting ease of sharing with the world, thanks just to one impartial recommendation or opinion. Only transact the goods, as bizarre sales locations are unique commerce grids: Social channels enable users to interact key consumers with professional purchasers [5]. Consumers have access to a substantial volume of material regarding procedures in the act of buying which are generated by partners. Existing customer behavior, user-generated content quantity, and diversity, along with the ultimate increase in electronic fans' dynamic possibility,

S. Singh · D. Kaushal · M. Singh · S. Taneja (✉)
Graphic Era Deemed to Be University, Dehradun, Uttarakhand 248002, India
e-mail: drsanjaytaneja1@gmail.com

V. Goar et al. (eds.), *Deep Learning and Visual Artificial Intelligence*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-97-4533-3_2

open up multiple business and research opportunities. Automatically handling software volumes present a technological challenge since the social web data quantity is growing rapidly. "Volume the volume, the subject of the volume of fans, the volume of heterogeneity." From there, a need for more advanced research tools keeps on increasing, which furnishes data worth derived from digital typewriters [6].

Deep learning's capability to decipher these singular intricacies of the knowledge and understanding of the phenomenon constituting eWOM is the primary reason for the current investigation on the marriage of eWOM and deep learning. Traditional methodologies of sentiment analysis and opinion mining prove to be very inefficient in constructing all of these nuances, contexts, and emotions associated with the numerous forms of UGI. A subset of AI known as deep learning was influenced by the human brain's structure and functionality. It "learns" by deconstructing the confusing eWOM with great ability and effectively spotting designs, contexts, and attitudes never seen before.

Considering the ever-increasing pace of consumer-generated content, organizations, marketers, and researchers should embrace the window of opportunity that deep learning enables to develop informed customer insights for actionable insights, in addition to improving their customer life cycle based on information learned from the insights [7]. This article therefore seeks to explore how electronic word-of-mouth has evolved, the challenges facing its analysis, and some of the deep learning-based methods of addressing the same. By synthesizing some of the current applications and methodologies, and offering projections for future trends in the niche area, we believe we will contribute to the extant body of knowledge regarding the convergence of eWOM and deep learning. It will also serve as a template for charting out future research goals in this dynamic area.

2 Evolution of eWOM

2.1 Historical Perspective

The first signs of eWOM can be traced back to the early days of the World Wide Web in the introduction of forums and discussion boards that allowed people to voice their thoughts and experiences. Since advertising and marketing were more restricted, talking topics were focused around various small communities where the knowledge exchange rate was as low as the exchange of that knowledge. The mid 2000s marked the acceleration of eWOM development. During this period, platforms such as Facebook, Twitter, and later on Instagram became the venue where consumers could voice their thoughts, recommendations, and complaints globally.

As a consequence, the eWOM reached critical mass for the first time in history and allowed people to express their opinions to online audiences of unprecedented size. Simultaneously with social media, organized review sites appeared as a venue for consumers to make detailed evaluation of products, services, experiences, and blog posts, tweets, photographs, ratings, reviews, messages, and videos. Web sites like Yelp, TripAdvisor, and Amazon Reviews took a major role in gathering and spreading eWOM in providing valuable information for potential consumers.

2.2 Impact on Consumer Behavior

The evolution of eWOM has dramatically impacted consumer behavior, disrupting well-established decision-making processes and redefining consumers' path to purchase [8]. Unlike traditional advertising, eWOM is more authentic and trustworthy because it is based on recommendations between peers rather than brand-pushed marketing assertions. Consumers increasingly consider the eWOM a pivotal source of information in the stage of pre-purchase [9]. The knowledge of crowds of diversified respondents provides a 360-degree view of the product and service, helping prospects in making well-informed decisions. Positive eWOM serves as an influential recommendation, which influences brand preference and customer loyalty, whereas negative eWOM may ruin a brand in a matter of hours.

Finally, it is the immediacy of eWOM and its viral nature that make the information spread so rapidly. Any review or recommendation, be it bad or good, may target huge audiences in a matter of seconds, which affects a brand's image and consumer attitudes immensely. In conclusion, the eWOM evolution process is like a summary of the digital progress, from the early forums to social media and review platforms [10, 11]. The direct impact on consumer behavior means that businesses should learn to control and manage eWOM and use the information obtained from analysis for strategic development and decision-making. The following sections will further investigate the difficulties of processing and analyzing eWOM and the role of deep learning in dealing with the vast amounts and variety of these data.

3 Challenges in Analyzing eWOM

Volume and Variety

The recent years of eWOM analysis proved to be much challenging since it was heavy on user-based data and voluminous. The review, image, and video uploads on online platforms are overwhelming, and traditional strategies of data mining are unable to find meaningful insight in this enormous amount of information [12, 13]. The scale challenge is prolonged by the fact that eWOM is rapid and never boring. Social media, forums, or review Web sites currently at all times generate new contents; therefore, online analytical tools must be nigh to a true real time as much as possible so as to stay up-to-date with the most current sentiment and trend. This large dynamic range implies deploying scalable, non-static solutions powerful enough to carry out activities on large data sets decisively and demandingly without running the risk to compromise quality.

Apart from the scale, varied forms on eWOM data delivery are a considerable analytical issue. Some utilize a great variety of emoticons in reviews along with text, while others use photographs or videos, which have additional meanings attached to it. Conventional sentiment analysis has consistently faced challenge with eWOM due to the richness of expressions and variants it delivers, indicating the need to invest in versatile methods which can cover different forms of data.

Sentiment Analysis

When it comes to online assessments, sentiment analysis is simply another name for machine studying. Sentiment evaluation is a key part of eWOM evaluation, but it is constrained by the truth that consumer reviews are complicated and context-reliant. Conventional sentiment evaluation fashions frequently rely on a lexicon-primarily based or rule-primarily based technique, which may also fight to perceive sarcasm or cultural peculiarities in user-generated content material. The binary classification of sentiment as positive or negative or neutral is too simplistic to reflect the complexity of consumer opinion.

Many eWOM expressions imply sentimentality, which means that one example of content material may have both positive and negative sentiments. Understanding the context is vital to determining genuine emotional expression, and this evaluation will require much more advanced analytical tools than conventional emotional evaluation strategies, and right now, emotional evaluation does not account for emotional development over time. Users may change their opinions and feelings based on other considerations. Then, there is the problem of the temporal view of emotions and the difficulty of selecting the right models which can capture the dynamics of emotional change, which adds even more complexity. Integration of deep learning techniques will be vital facing those challenges.

Because they can automatically learn and symbolize functions, deep studying fashions in determining on eWOM with the large amount, variety, and complexity of data seem like a promising approach. The following subsections will explore the specific approaches and methodologies to which deep gaining knowledge can contribute meaningful understanding of the eWOM complicated landscape and facilitate overcoming the challenges.

4 Deep Learning in eWOM Analysis

Natural Language Processing (NLP) in eWOM

In natural language processing, one of the most important tasks is text classification. In recent years, many data-driven approaches to text classification have been offered. Recent research, however, has focused on deep learning-based methods which have demonstrated outstanding success on a variety of text classification tasks. NLP is a field that empowers computer systems to communicate, interpret, and develop human-like text comprehension. Sentiment analysis is a branch of NLP that uses similar methodologies and technologies to identify different forms of human emotional feedback on a product, service, or establishment. The development of social media, online reviews, and other social and multimedia sharing services have resulted in wide availability to large amounts of digital text information.

The rapid progress of deep learning is driven by the demands of eWOM analysis; today, sentiment analysis is the most active NLP topic. User opinions are determined by their interaction with products, services, or environments, and opinions influence other people's decisions. The ability of chronic neural networks to understand eWOM text is exceptional. This entails understanding that one event takes place one after another and that context affects the creation of eWOM. Previous machine learning techniques' disadvantages in certain ways limited data classification and generalization capacity, shallow-structured algorithms that were difficult and expensive to train, and an inability to model complex functions. Deep learning techniques aim to correct the gaps in current methods: for example, attention mechanisms aid shrink the performance shortfall. Attention mechanisms are used to fuel interpretable deep learning applications. Due to attention mechanisms, models can perform text analysis and classify them based on the input.

Image and Multimedia Analysis

Another key transformation present in the evolution of eWOM is the integration of images, videos, and other multimedia elements by users as part of their expression. With such a wide variety of data types to analyze, traditional text-based analysis methods cannot keep up with the demands. Deep learning, as a technique that excels in analyzing images and multimedia data, can substantially improve the analysis of eWOM by considering the content associated with it more holistically. Convolutional neural networks are well known for their capabilities in analyzing images. This technology can be used to extract features from the images shared as part of eWOM. For instance, when working with product reviews, images may reflect the characteristics of the product, possible product use cases, or even the user experience. CNNs can automatically learn the important features of these visuals, contributing data to the analysis process. Moreover, multimodal deep learning models that work with textual and visual data at the same time can also be used to enhance the variety of the analysis. These models can also account for the correlation between sentiments expressed through texts and the context reflected in images. The adaptability of deep learning methods to multiple data types makes them well suited to address the challenges presented by eWOM. Deep learning that integrates text and visual data can bring more sophisticated and informed ways to analyze the multiple layers of the data presented in the electronic word-of-mouth. The following sections provide more information about the application of deep learning to eWOM analysis.