Machine Learning-Based Malware Classification in Real-Time IoT Scenarios

Arslan Rafi
Independent Researcher
Email: arslanrafi@ymail.com

Attaullah Buriro
School of Computer Science and Electronic Engineering
University of Essex, Colchester, UK
Email: attaullah.buriro@essex.ac.uk

Muhammad Azfar Yaqub and Antonio Liotta
Faculty of Engineering
Free University of Bozen-Bolzano
Bolzano 39100, Italy
Email: {myaqub, antonio.liotta}@unibz.it

***Abstract*—Ensuring the security of future network infrastructures, which include 5G/6G, Internet of Things (IoT) and Intelligent Transportation Systems (ITS), requires precise identification and detection of malicious/malware families. Although current malware identification methods achieve high accuracy, their validation is largly confined to a narrow range of malware families/samples. Analyses often prioritise families with abundant samples, introducing potential bias and limiting representativeness in classification results. This leads to unreliable detection in real-world heterogeneous network environments. This study addresses the gap by improving malware identification accuracy and robustness through an analysis of dataset size, and class balance, the effect of temporal data augmentation on classifier performance. The study demonstrates that maintaining balanced sample sizes across various malware families significantly improves classifier accuracy by mitigating bias towards majority classes. Our approach leverages state-of-the-art classifiers and two augmentation techniques, Synthetic Data Vault and Synthetic Minority Over-sampling Technique, to enhance the malware classification, particularly in vulnerable enviornments such as edge networks, ITS and IoT.**

***Index Terms*—Malware Detection, Generative Adversarial Networks, Deep Neural Network, Convolutional Neural Network**

I. Introduction

In today's interconnected world, the security of modern network infrastructures—including next-generation technologies such as 5G/6G networks and IoT—is of paramount importance for protecting sensitive information and ensuring the continuity of digital operations. These environments, due to their distributed nature and massive scale, are increasingly vulnerable to sophisticated malware attacks that can exploit network and device-level weaknesses [1]. With the integration of IoT and 5G/6G technologies, ITS are evolving into connected and autonomous ecosystems that introduce new security challenges. Real-time data exchange among vehicles, roadside units, and cloud platforms increases the risk of malware propagation [2]. The high mobility and dynamic topology demand adaptive, low-latency classification methods [3].

Malware, encompassing a wide array of malicious software, pose significant threats to network security by exploiting vulnerabilities and compromising systems [4]. As cyber threats continue to evolve in complexity and sophistication, it has become a dire need to develop and refine malware classification systems that can effectively identify and mitigate these threats [5]. Traditional malware analysis methods, e.g., signature-based [6] or behavior-based [7], rely on predefined patterns or manual analysis of malware characteristics or behaviors. However, these methods have proven ineffective against new or unknown malware, as they are unable to recognize malware that does not match the existing patterns or profiles.

Machine Learning (ML) methods offer a powerful alternative to traditional malware detection techniques by detecting malware through data-driven approaches that can identify complex patterns without requiring prior knowledge or human intervention [8]. One of the key advantages of ML techniques is their adaptability to the evolving nature of malware, allowing them to continuously improve their performance by learning from new data and feedback [9]. However, the effectiveness of ML models can be significantly influenced by quality and balance of training data. Imbalanced datasets, where certain classes are underrepresented, often lead to biased classifiers that disproportionately favor the majority classes [10]. To mitigate this challenge, achieving a balanced class distribution is essential for enhancing the system's capacity to accurately identify and classify malware. This is essential for developing robust defense mechanisms [11].

In this paper, we propose a ML-based framwork to enhance the reliability of malware categorization systems considering higly skewed malware classes. Technically speaking, we demonstrate the efficacy of balanced class distributions and effective data augmentation using Synthetic Data Vault (SDV) and Synthetic Minority Over-sampling Technique (SMOTE), towards the development of a reliable and accurate malware classification systems. This study exploits maximum malware families (49, in total) taken from Lu et al. [12], each containing at least 100 training samples and reports the highest accuracy of 92.54%, 93.26% and 95.21%, achieved by Random Forest (RF) classifier.

In summary, the major contributions of this study are the following:

- Comprehensive analysis of how class balance using SDV and SMOTE-based affects the performance of malware classification systems.
- Demonstration of significant improvement in classifier accuracy by equalizing the number of samples across

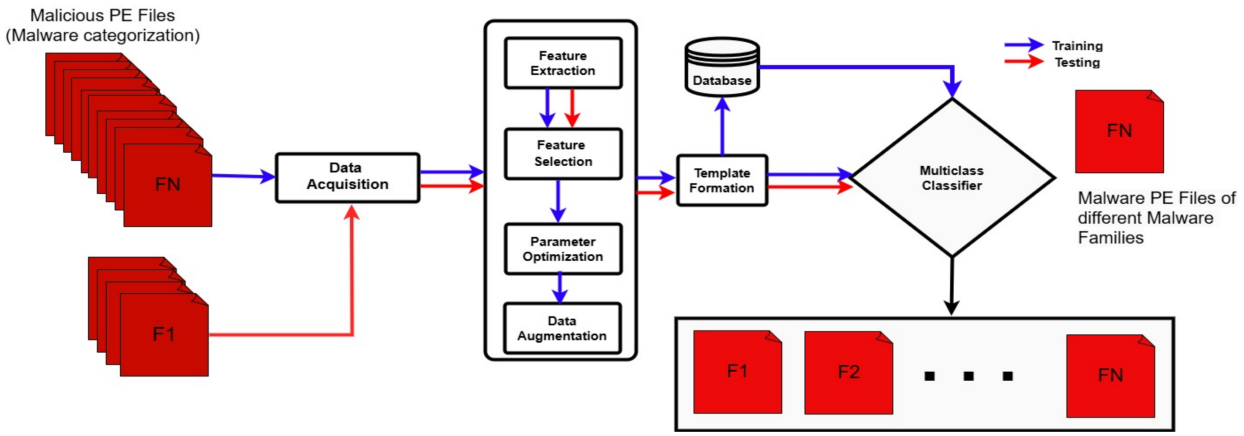


Fig. 1: Our approach of enhancing malware categorization accuracy.

different malware families.

- Statistical analysis confirming that RF yields significantly better results on SMOTE-augmented data compared to the original imbalanced dataset.

II. Literature Review

The impact of class balance on the accuracy of classifiers in malware analysis has been extensively studied, particularly in the context of modern, real-time network environments such as IoT and 5G/6G infrastructures. Wang et al. [13] reported that classification error decreased with increasing training data size, achieving a malware detection accuracy of up to 98.7%. This study highlights the importance of having a balanced dataset to improve classifier performance. Similarly, Alzammam et al. [14] demonstrated that methods such as oversampling can positively affect classification performance. The author's comparative analysis of imbalanced multi-class classification using Convolutional Neural Networks (CNN) demonstrates substantial accuracy gains once class imbalance was addressed. Extending these insights to next-generation networks, recent research has adapted balancing techniques for the real-time constraints and heterogeneity of IoT and 5G/6G systems. For instance, Chen and Ye [15] utilized hybrid resampling with ensemble models like gcForest on the highly imbalanced IoT-23 dataset, significantly boosting malware detection accuracy in edge environments.

Class imbalance, where some classes have significantly more samples than others, can lead to biased classifiers that favor the majority class. Equalizing the number of samples across classes helps mitigate the bias towards the majority class. Techniques such as oversampling (e.g., SMOTE [16]) and data augmentation (e.g., GANs) generate synthetic samples for minority classes, providing a more balanced dataset. This balance allows classifiers to learn more effectively from all classes, leading to improved accuracy and reliability in malware detection.

The effectiveness of various data augmentation and oversampling techniques in malware classification has been extensively studied. Burks et al. [17] found that incorporating synthetic malware samples generated by Variational Autoencoders (VAE) into the training data improved the accuracy of a ResNet-18 classifier by 2%. This study highlights the potential of generative models in enhancing classifier performance by providing additional training samples that mimic real malware. Overall, these studies highlight the importance of data augmentation and oversampling techniques in improving the accuracy and reliability of malware classification systems, however, most fail to apply these methods effectively on larger and more diverse datasets. Our work further integrates temporal analysis to capture the evolution of malware behavior, an aspect rarely addressed in prior studies.

III. Our Approach

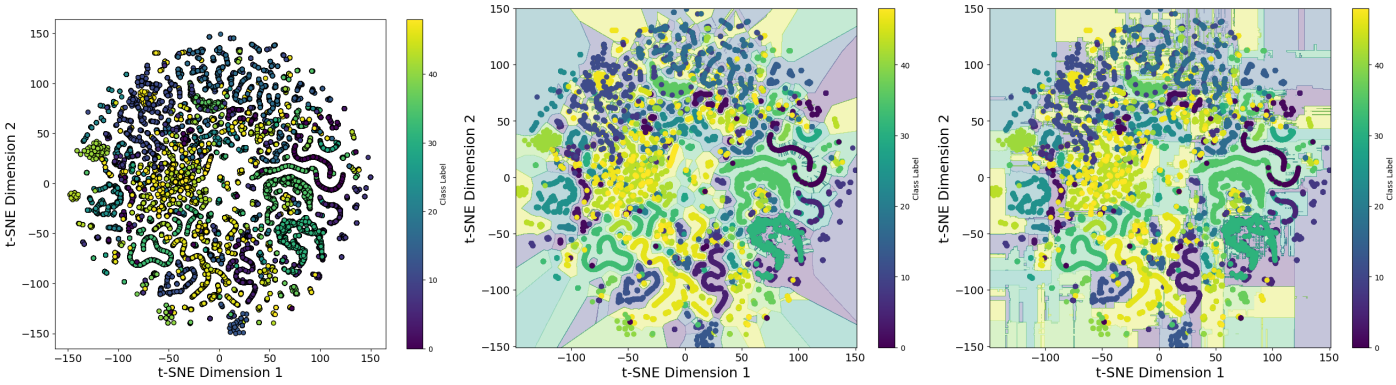
The proposed malware categorization process begins with acquiring Portable Executable (PE) files from Blue Hexagon Open Dataset for Malware AnalysiS (BODMAS) [18], which includes both benign and malicious files, i.e., malware. Our approach involves extracting features using the Library to Instrument Executable Formats[1] (LIEF) library, followed by feature selection to reduce the initial feature vector from 2381 features to a more manageable 25-feature vector. Subsequently, we perform oversampling to enhance samples from minority classes, ensuring that the classifier's decisions are not biased towards majority classes. The optimized classifiers are then tested on the same extracted and selected features from unseen malware samples to determine the malware family the query sample belongs to (see Figure 1). The performance of the classifier is evaluated both before and after oversampling to assess the effectiveness of synthetic data generation using SDV and SMOTE.

IV. Methodology

A. Dataset

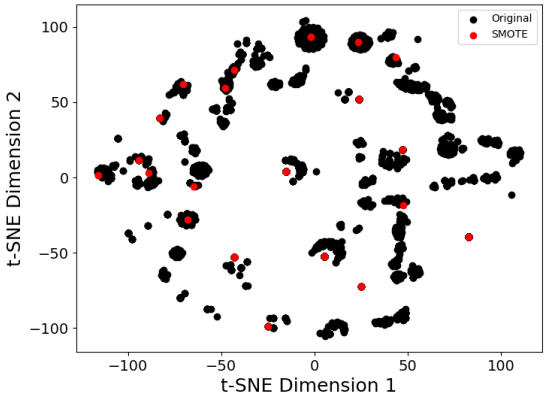
We utilized the BODMAS dataset [18] to evaluate our methodology. This dataset encompasses a substantial collection of 57,293 malware samples from 581 distinct families and 77,142 benign files, compiled between August 2019 and

[1] https://github.com/lief-project/LIEF

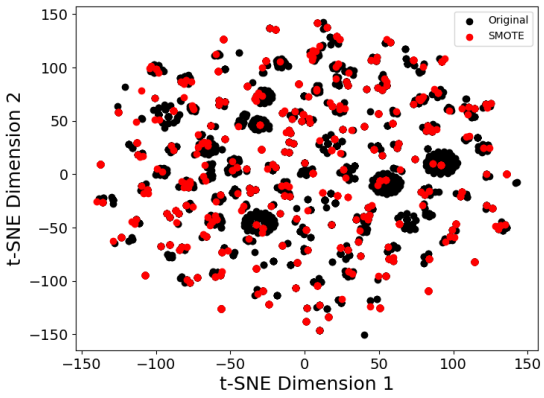


(a) (b) (c)

Fig. 2: t-SNE representation and classification boundaries of different classifiers (a) Original, (b) KNN & (c) RF.



(a)



(b)

Fig. 3: Visual representation of SMOTE-based data augmentation for some classes (a) Wacatac and (b) upatre. Due to space limitations, we show these illustrations for two selected classes only.

September 2020. The dataset includes disarmed malware binaries, feature vectors, and metadata. Each sample or observation is represented by a 2381-feature vector, which is labeled as either benign or malicious, with additional metadata detailing the specific malware family. We employed the LIEF Library to extract features from executable files in our study. The same feature set, originally extracted by the creators of the datasets in [18] [19].

B. Feature Subset Selection

Feature subset selection involves identifying the most effective subset of features that improve classification accuracy while simplifing the learning process for the classifier [20]. To perform transparent evaluation, we chose to exploit Sequential Forward Selection[2] (SFS) features previously computed in the study [21] as this approach helps identify the most relevant features for our classification task without requiring model retraining on high-dimensional transformations. Table I presents the selected features used for malware categorization (multi-class classification).

C. Classifiers Selection

Classifiers are core ML models that learn from data and assign labels to new samples. Their performance varies with the dataset and task, making selection crucial. In our research, we employ two effective yet interpretable ML classifiers (K-Nearest Neighbor (KNN) and Random Forest (RF)) due to their proven performance in prior studies [21].

To visualize the classifier behavior, t-Distributed Stochastic Neighbor Embedding (t-SNE) [22], is used to project high-dimensional features into a two dimensional space, reveling class separability. The decision boundaries depicted in the t-SNE plots (see Figure 2) illustrate how well each classifier distinguishes between classes, with distinct clusters indicating robust pattern recognition and reliable discrimination.

D. Analysis

Temporal data splitting is essential for evlauating classifiers in dynamic enviornments. It simulates real-world conditions where model encounters evolving and unseen threats, thus preventing optimistic performance estimates, and enhance generalizability.

In this study, we focused on the top 49 malware classes. The smallest family, Glupteba, contains 105 samples, while

[2] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SequentialFeatureSelector.html

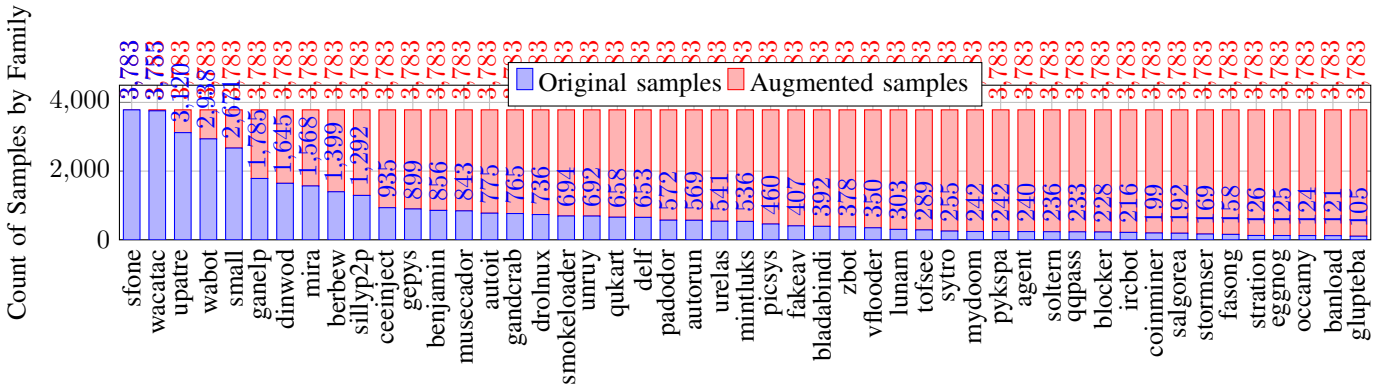


Fig. 4: The distribution of original and generated samples for 49 families (having more than 100 samples).

TABLE I: SELECTED SFSS FEATURES FOR MALWARE CATEGORIZATION [21].

Settings	F#1	F#2	F#3	F#4	F#5	F#6	F#7	F#8	F#9	F#10	F#11	F#12	F#13	F#14	F#15	F#16	F#17	F#18	F#19	F#20	F#21	F#22	F#23	F#24	F#25
Categorization	323	339	617	626	672	685	722	734	742	745	836	866	1031	1060	1168	1282	1329	1412	1608	1653	1720	2083	2119	2251	2354

the largest, Sfone, has 3,783. The remaining families each contain a number of samples between these two. For model training, we used the first 80% of the data chronologically, reserving the remaining 20% as a test set. This test set remained completely unseen during feature selection, parameter optimization, and synthetic data generation, ensuring that the model's performance metrics are a true reflection of its capability to handle new, previously unseen malware. By adopting this temporal data split, we not only safeguard the integrity of our model evaluation but also enhance the model's relevance and applicability to real-world scenarios.

E. Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE is a powerful data augmentation method designed to tackle class imbalance in machine learning datasets. Introduced by Chawla et al [16], it addresses the challenge of imbalanced classification datasets, where the minority class has too few samples for effective learning. The goal is to improve classifier performance on the minority class(es). This process involves identifying minority class samples, selecting their nearest neighbors, and creating new synthetic samples along the line segments connecting the samples and their neighbors.

As shown in Figure 4, we used *sfone* as a reference to standardize the sample sizes across different classes. Specifically, we aimed to partially increase the number of samples for each family but did not observe any significant improvement. Finally, we balanced all families to match the total number of sfone samples (3,783) to ensure equal class representation during model training. The figure illustrates the class balancing process: blue bars indicate the original sample count, while the red bars show the synthetic samples added to reach 3783 instance per class. This visualization clearly demonstrats the initial imbalance and the the effect of augmentation in achieving uniform class representation.

Our synthetic data generated using SMOTE closely resembles the original samples, highlighting the effectiveness of the augmentation process. As shown in Figure 3, the similarities between the original and synthetic data are evident. The distribution patterns of both types of samples appear nearly identical, suggesting that SMOTE has successfully captured and replicated the underlying structure of the original data. Due to space limitations, we present these illustrations for only two classes, but similar trends were observed across other classes as well.

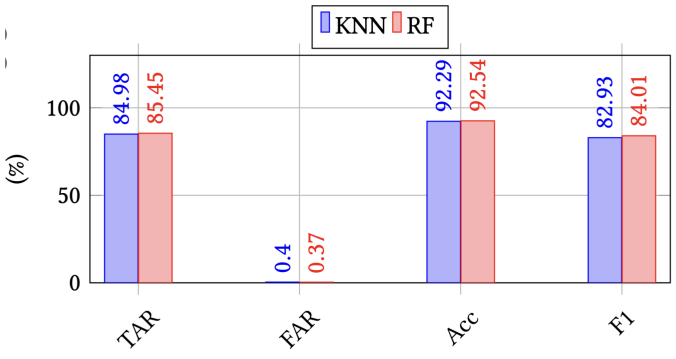
F. Synthetic Data Vault (SDV)

SDV [23] is an advanced data generation framework designed to create synthetic data that closely resembles real-world datasets. Developed by the MIT Data To AI Lab, SDV aims to address various challenges in data science, including data privacy, data sharing, and class imbalance. By generating high-quality synthetic data, SDV enables researchers and practitioners to perform robust analyses without compromising sensitive information.

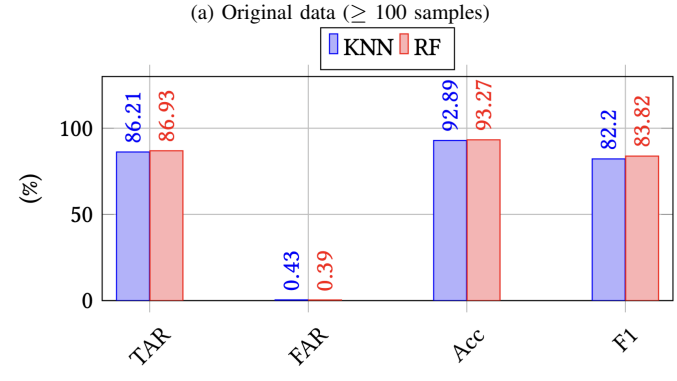
The process involves training generative models on real datasets to learn their underlying patterns and distributions. Once trained, these models can generate new synthetic samples that mimic the statistical properties of the original data. This approach is particularly useful in scenarios where access to real data is limited or restricted due to privacy concerns.

V. RESULTS

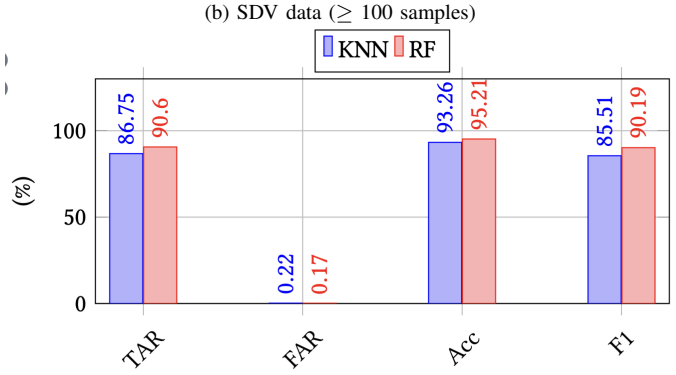
In the context of malware categorization, where the goal is to classify samples into specific malware categories, several important metrics are used to evaluate classifier performance. The True Positive Rate (TPR)/True Accept Rate (TAR) measures the proportion of samples that are correctly identified as belonging to their respective malware category. Conversely, the False Negative Rate (FNR)/False Reject Rate (FRR) represents



(a) Original data ($\geq$ 100 samples)



(b) SDV data ($\geq$ 100 samples)



(c) SMOTE augmented data ($\geq$ 100 samples)

Fig. 5: Comparison of classifiers performance on Original (5a), SMOTE augmented (5b, samples for 49 malware families, respectively.

the proportion of samples that are incorrectly classified as not belonging to their true category. Similarly, the False Positive Rate (FPR), or False Accept Rate (FAR), measures the proportion of samples that are incorrectly classified into a malware category that they do not belong to. The True Negative Rate (TNR), or True Reject Rate (TRR), captures the proportion of samples correctly identified as not belonging to a specific malware category.

For our analysis, we focus on reporting the TAR and FAR, overall accuracy and F1 score obtained by the classifier. By presenting these metrics, we provide a concise and relevant evaluation of the classifier's ability to accurately categorize malware samples. The results, detailed in Figure 5, highlight the effectiveness of our chosen classifiers in distinguishing between different malware categories. We do not report FRR (as FRR = 1 - TAR) and TRR (TRR = 1 - FAR) to avoid redundancy.

Figures 5a, 5b and 5c summarise our average results for 49 malware families. Recall that 49 classes contained $\geq$ 100. It is worth noting that our original data, which is highly skewed (containing as many as 3783 samples and as few as 105) resulted in comparatively lower accuracy, i.e., we report 84.98% and 85.45% TAR at just 0.4% and 0.37% FAR yielding an overall accuracy of 92.29% and 92.54% for KNN and RF, respectively. Figure 5 also depicts the F1 score, which seems quite acceptable given the data skewness.

Our results also demonstrate the efficacy of our SMOTE-based synthetic data augmentation scheme. For example, compared to the classifiers performance on original (see figure 5a), and SDV augmented train set (see Figure 5b), SMOTE augmented (see Figure 5c), dataset yielded significantly higher accuracy. Generally speaking we observed an upward trend in all the success parameters, TAR, accuracy, F1 score and downword trend in FRR and FAR. We report a TAR of 86.75% and 90.6%, at just 0.22% and 0.17% FAR, and overall accuracy of 93.26% and 95.21% for KNN and RF classifiers, respectively.

In summary, the accuracy rose to 93.26% for KNN and to 95.21% for RF when data balancing schemes are employed to substantiate the statement that classification is best with balanced data. SMOTE outperformed SDV by effectively managing class unbalance with targeted synthetic sample construction to yield stabler and more accurate outputs.

To statistically evaluate differences in classifier performance across experimental settings, we perform T-Test. A T-Test assesses whether the observed difference between the means of two groups is statistical significant, thereby indicating whether the variation is attributed to random chance or reflects a true underlying effect. It calculates a p-value, which reflects the probability that the difference between groups occurred by chance, with a lower p-value (typically less than 0.05) indicating statistical significance. Applying this method to compare the performance of KNN and RF classifiers on the original, SDV-augmented and SMOTE-augmented datasets reveals important insights. The T-Test results for the original and SDV dataset indicated no statistically significant difference between the two classifiers, suggesting that KNN and RF performed similarly on these two datasets. However, the results shift when evaluating the SMOTE-augmented data: RF significantly outperformed KNN, as highlighted by a P-value of 0.00569182127359239.

The superior performance of the RF model can be attributed to its ensemble architecture, which enables it to capture and model complex data patterns more effectively. In contrast, KNN's reliance on distance metrics may hinder its ability to align with the data's underlying structure, particularly in the original dataset, leading to its relatively weaker performance. However, when comparing the performance of RF across different datasets, we found no statistical difference between the original and SDV-augmented datasets. In con-

trast, there was a statistically significant difference between the original and SMOTE-augmented datasets (P-value of 0.000677719669731569) and between the SDV and SMOTE-augmented datasets (P-value of 0.0006749946171746531). This suggests that RF achieved significantly better results when trained on the SMOTE-augmented dataset.

VI. Conclusions and Future Work

This study introduces a real-time IoT malware detection framework leveraging deep learning, with special focus on class-balance mitigation and precision enhancement. Data augmentation strategies such as SMOTE and SDV improve balance in the BODMAS data set with 95.21% precision using Random Forest. These strategies handle class imbalance and develop robust models for various malware, with quality data playing a significant role in network security.

Future work will involve the integration of advanced deep learning models and hybrid approaches to further enhance malware detection accuracy. Additionally, investigating the impact of real-time data streams and concept drift on classifier performance could provide valuable insights. Expanding the dataset with more diverse and recent malware samples, together with investigating novel data augmentation techniques, would further enhance the robustness and reliability of malware detection systems.

VII. Acknowledgment

This work was carried out in the context of project SELF4COOP (Self-optimizing Networked Edge Control for Cooperative Vehicle Autonomy), funded by the European Union under Next Generation EU, Mission 4 Component 2 - CUP E53D23000910001. Views and opinions expressed are those of the authors only and do not necessarily reflect those of the EU or the EU REA. Neither the EU nor the granting authority can be held responsible for them.

References

[1] M. R. Jeebodh and N. Baliyan, "Iot malware detection using deep learning," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–6, IEEE, 2024.

[2] P. Upadhyay, S. Goyal, V. Marriboyina, and S. Kumar, "Securing vehicular internet of things (v-iot) communication in smart vanet infrastructure using multi-layered communication framework and novel threat detection algorithm," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 6s, pp. 789–803, 2024.

[3] H. A. El Zouka, "An efficient and secure vehicular networks based on iot and cloud computing," *SN Computer Science*, vol. 3, no. 3, p. 240, 2022.

[4] S. J. Stolfo, S. Hershkop, K. Wang, O. Nimeskern, and C.-W. Hu, "A behavior-based approach to securing email systems," in *Computer Network Security: Second International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2003, St. Petersburg, Russia, September 21-23, 2003. Proceedings 2*, pp. 57–81, Springer, 2003.

[5] M. Sikorski and A. Honig, *Practical malware analysis: the hands-on guide to dissecting malicious software*. no starch press, 2012.

[6] "Malware detection top techniques today." [Accessed: Dec. 22, 2023].

[7] H. S. Galal, Y. B. Mahdy, and M. A. Atiea, "Behavior-based features model for malware detection," *Journal of Computer Virology and Hacking Techniques*, vol. 12, pp. 59–67, 2016.

[8] V. Dhingra, J. Singh, and P. Kaur, "Detecting and analyzing malware using machine learning classifiers," in *International Conference on Next Generation Systems and Networks*, pp. 197–207, Springer, 2022.

[9] M. Aslam, D. Ye, M. Hanif, and M. Asad, "Adaptive machine learning: A framework for active malware detection," in *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, pp. 57–64, IEEE, 2020.

[10] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[11] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent data analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[12] Q. Lu, H. Zhang, H. Kinawi, and D. Niu, "Self-attentive models for real-time malware classification," *IEEE Access*, vol. 10, pp. 95970–95985, 2022.

[13] P. Wang and Y.-S. Wang, "Malware behavioural detection and vaccine development by using a support vector model classifier," *Journal of Computer and System Sciences*, vol. 81, no. 6, pp. 1012–1026, 2015.

[14] A. Alzammam, H. Binsalleeh, B. AsSadhan, K. G. Kyriakopoulos, and S. Lambotharan, "Comparative analysis on imbalanced multi-class classification for malware samples using cnn," in *2019 International Conference on Advances in the Emerging Computing Technologies (AECT)*, pp. 1–6, IEEE, 2020.

[15] J. Chen and R. Ye, "Network threat detection: Addressing class imbalanced data with deep forest," *arXiv preprint arXiv:2506.08383*, 2025.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[17] R. Burks, K. A. Islam, Y. Lu, and J. Li, "Data augmentation with generative models for improved malware detection: A comparative study," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 0660–0665, IEEE, 2019.

[18] L. Yang, A. Ciptadi, I. Laziuk, A. Ahmadzadeh, and G. Wang, "Bodmas: An open dataset for learning based temporal analysis of PE malware," in *2021 IEEE Security and Privacy Workshops (SPW)*, pp. 78–84, IEEE, 2021.

[19] H. S. Anderson and P. Roth, "Ember: an open dataset for training static pe malware machine learning models," *arXiv preprint arXiv:1804.04637*, 2018.

[20] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[21] A. Buriro, A. B. Buriro, T. Ahmad, S. Buriro, and S. Ullah, "MalwD&C: a quick and accurate machine learning-based approach for malware detection and categorization," *Applied Sciences*, vol. 13, no. 4, p. 2508, 2023.

[22] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[23] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," in *2016 IEEE international conference on data science and advanced analytics (DSAA)*, pp. 399–410, IEEE, 2016.