Bima Journal of Science and Technology, Vol. 9(1A) Mar, 2025 ISSN: 2536-6041

DOI: 10.56892/bima.v9i1A.1259

A Hybrid Shallow CNN-LSTM Model for Enhanced Malware Classification

Sani Aliyu[1]*, Ahmed Abubakar Aliyu[2], Muhammad Aminu Ahmad[2], Mohammed Ibrahim[2],
Saadatu Abdulkadir[1] and Abubakar Mu'azu Ahmed[1]

[1]Department of Informatics, Kaduna State University, Nigeria
[2]Department of Secure Computing, Kaduna State University, Nigeria

Corresponding Author: sanialiyu@gmail.com

ABSTRACT

The cybersecurity field faces significant challenges in detecting and classifying malware due to the continuous propagation of more complex and varied malware. Imbalanced datasets where benign samples outweigh malware samples, which create significant problems within this domain resulting in biased model performance. This study developed a hybrid shallow CNN with LSTM integration to overcome the problem of malware classification. This method combines CNN's spatial feature extraction capabilities with LSTM's sequential pattern recognition to analyze malware's static and dynamic properties effectively. The methodology involved evaluating the model on a dataset with significant class imbalances through the adoption of SMOTE. The model evaluation relied on several key performance metrics, including accuracy, precision, recall, and the F1-score. The study also compared the developed model with existing models to demonstrate its superior performance The hybrid CNN-LSTM model proved effective by obtaining 99.44% accuracy for balanced datasets while sustaining 99.22% accuracy for imbalanced datasets. The results confirm that reducing class imbalance improves the accuracy of machine learning models for malware classification. The model demonstrated better performance than earlier research works, as shown through its higher accuracy rate. This research develops malware detection techniques and delivers a reliable solution for real-world cybersecurity applications.

Keywords: Convolutional Neural Network, Cyber Security, Malware, Long-Short-Term Memory.

INTRODUCTION

In today's digital landscape, the advent of malicious software attacks is a significant threat to cybersecurity (Alzaylaee et al., 2020). Malware or malicious software includes a variety of harmful executable software like viruses, worms, trojan horses, bots and spywares which are harmful to computer systems thereby causing significant damages (Krishnamurthi et al., 2022).

The diversity of malware coupled with its continuous growth is a significant challenge to network security (Abdullah et al., 2022). This massive growth was intensified by the COVID-19 pandemic where online activity massively increased leading to heavy reliance on network infrastructure which led to the risk of malevolent actors targeting the infrastructure for malware attacks (Hakak et al., 2020).

This massive growth led to an increase of up to 360,000 new malware variants detected in 2020 representing a 5.2% increase from previous years. Automated malware creation tools like SpyEye and Zeus have significantly contributed to the scourge (Awan et al., 2021).

Other devices connected to the Internet like IoT devices are also at high risk of malware attacks potentially leading to data leaks, unauthorized access and physical injury,

DOI: 10.56892/bima.v9i1A.1259

financial losses etc. (Awan et al., 2021). Therefore, robust cybersecurity measures are necessary to protect the millions of IoT users from these harmful assaults.

Despite efforts by cybersecurity providers and antivirus software manufacturers to identify and block malware, a significant number of samples, including "zero-day" malware, evade conventional scanning tools reliant on signatures (Hindy et al., 2020). As a result, the information security industry is reevaluating malware recognition techniques, moving beyond signature-based models.

While the cybersecurity industry continuously strives to monitor and combat malware, cyber attackers persist in developing evasive techniques such as polymorphism, metamorphism, and code obfuscation, outpacing traditional mitigation systems (Imamverdiyev & Baghirov, 2024). The emergence of these techniques, coupled with the increasing number of households with multiple vulnerable devices, necessitates the development of fast and reliable techniques to identify and combat new malware (Awan et al., 2021).

Malware classification plays a fundamental role in malware analysis as it helps in understanding the diverse categories of malwares, their potential impact on personal computers, and the necessary defense strategies. When a malicious software is detected on network traffic, it becomes necessary to properly assign it to the appropriate malware family through a classification mechanism. While several methods exist for detecting known malware, identifying zero-day malware remains a challenging endeavor in the field. Building a reliable malware classifier becomes especially challenging due to the scarcity of high-quality labeled data (Di Troia et al., 2019).

Several researchers have explored deep learning-based techniques to enhance malware classification. Karat et al. (2024) introduced a CNN-LSTM hybrid model that integrates deep learning and behavioural analysis to improve malware detection accuracy. Their study demonstrated that traditional static signature-based approaches are inadequate for identifying advanced malware variants, while CNNs and LSTMs together effectively classify malware based on API call sequences. The model achieved 96% validation accuracy, confirming the effectiveness of deep learning-based malware classification (Karat et al., 2024). However, their study did not address dataset imbalance, which may impact the model's generalisation capabilities.

Similarly, Bensaoud and Kalita (2024) proposed a CNN-LSTM-based malware classification model that incorporates API calls and opcode sequences as feature representations. The researchers experimented with n-gram representations (N = 2, 3, 10) and fine-tuned various deep learning architectures, including ConvNeXt, RegNetY, Swin-T, and EfficientNetV2, achieving 99.91% accuracy using 8-gram sequences (Bensaouda & Kalita, 2024). Their study demonstrated that integrating NLP-based techniques such as TF-IDF and Bag-of-Words (BoW) can enhance malware classification performance. However, despite the high accuracy reported, their work did not explore the impact of adversarial attacks or model robustness against evolving malware threats, which remains a significant challenge in real-world deployment.

The problem addressed in this study is that of data class imbalance through the adoption of SMOTE. The research by Karat et al. (2024) implemented a CNN-LSTM hybrid model for malware detection, achieving validation accuracy of 96%. However, the study did not address the issue of data class imbalance. This study aims to utilize a one-dimensional CNN with LSTM and application of SMOTE to address the class imbalance.

While the study by Karat et al. (2024) presents a CNN-LSTM hybrid model for malware detection, achieving a high validation accuracy of 96%, it does not explicitly address the issue of class imbalance within the dataset. The dataset used in their research comprises 2,500 malware samples and 1,000 benign samples, indicating a skewed distribution that could potentially bias the model toward classifying malware more accurately than benign files. Imbalanced datasets in deep learning can lead to overfitting toward the majority class, reducing the model's ability to correctly identify minority-class instances. However, the study does not mention any application of data balancing techniques, such as the Synthetic Minority Oversampling Technique (SMOTE), which has been proven effective in mitigating class imbalance issues by generating synthetic samples for the under-represented class.

Based on the gap identified, this study will implement SMOTE in a CNN-LSTM-based malware detection system, which is expected to enhance the model's ability to generalise, reducing false negatives and improving the detection of previously unseen malware samples. This research gap presents an opportunity to develop an improved CNN-LSTM model that integrates SMOTE for data balancing, leading to a more robust and fair malware classification system.

MATERIALS AND METHODS

The flow of methodology for this research is shown in Figure 1, which outlines the steps undertaken to train and evaluate the performance of the model using metrics such as accuracy, precision, recall and f1-score of the hybrid shallow CNN-LSTM model. The implementation begins with the preprocessing of the selected dataset followed by model design through the selection of the appropriate parameters for the CNN component of the model, then training of the dataset and then passing the output to the LSTM classifier for accurate malware classification. Furthermore, the performance accuracy of the model will then be evaluated.

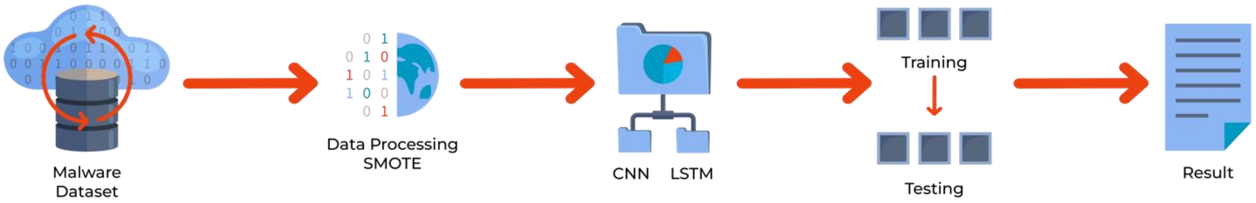

Figure 1: Methodology flow.

Dataset Acquisition and Description

This study used a publicly available malware dataset published by the United States Air Force for the purpose of research. The dataset comprises of 42,797 malware and 1,079 of benign data. The dataset is made up of API call sequences with each representing the first 100 unique and continuous API calls linked to the parent processes. (Bensaoud & Kalita 2024).

These sequences are extracted from the 'call' components of the Cuckoo Sandbox reports. Figure 2 shows the class distribution of the dataset.

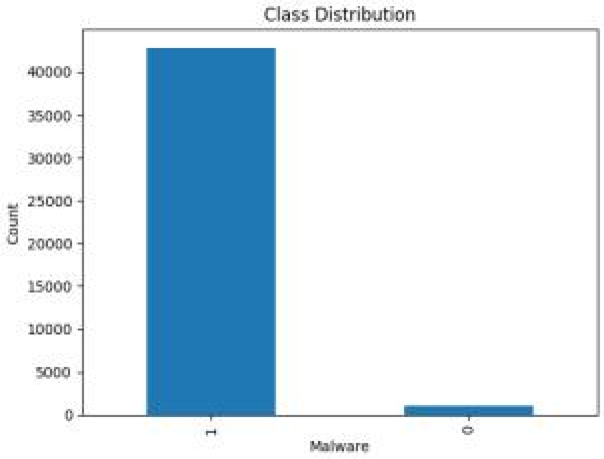

Figure 2: Data class distribution.

It is crucial to observe that the dataset depicted in Figure 2 shows a significant class imbalance, highlighting the need to employ appropriate balancing methods before directly inputting it into the model.

Dataset Distribution

The dataset for this research contains a total of 43,876 API call sequences which is broken down as 42,797 malware sequences and 1,079 benign sequences. With this significant disparity between the 97.5% malware and 2.5% benign classes, the need for careful handling of the model is imperative.

Before the training begins, the dataset will be separated into training and testing sets in the ratio of 75:25 for the preparation of the model.

The division into training and testing sets is essential for assessing the model's efficacy. The training set supplies essential data for the model to identify patterns and characteristics linked to both benign and malicious API calls, whereas the testing set facilitates an impartial evaluation of the model’s efficacy on unfamiliar data. This method aids in reducing overfitting and guarantees that the model can generalize effectively to new instances of malware and benign software, thereby improving its robustness in practical applications. Table 1 summarizes the distribution of the dataset.

Table 1: Dataset Distribution.

Category	Total Sequences	Training (75%)	Testing (25%)
Malicious API Calls	42,797	32,111	10,686
Benign API Calls	1,079	796	283
Total Sequences	43,876	32,907	10,969

Data Preprocessing

The collected datasets underwent preprocessing to convert them into suitable file formats that are compatible with the programming language used for model development. Additionally, a feature selection procedure was applied to the dataset, eliminating irrelevant attributes. This process improved the accuracy of the classifiers and optimized the processing time.

Proposed CNN-LSTM Model

After data preprocessing, the proposed model is then designed, adopting the following hyperparameters:

i. Input Shape: (X_train.shape[1], 1), where X_train.shape[1] represents the number of features, ensuring compatibility with the CNN-LSTM architecture.

ii. Convolutional Layer: 64 filters with a kernel size of 3, using ReLU activation. This layer extracts meaningful patterns from API sequences.

iii. MaxPooling Layer: Pool size of 2 to reduce spatial dimensions and computational complexity.

iv. LSTM Layers:

a. First LSTM layer with 100 units, returning sequences to retain temporal dependencies.

b. Second LSTM layer with 100 units, outputting a single vector for classification.

v. Dropout Layers: A dropout rate of 0.5 is applied after each LSTM layer to mitigate overfitting.

vi. Dense Layer: A single neuron with a sigmoid activation function for binary classification (malware or benign).

vii. Optimizer: Adam, chosen for its adaptive learning rate and efficient handling of sparse gradients.

viii. Loss Function: Binary crossentropy, suitable for binary classification tasks.

ix. Metrics: Accuracy is used as the primary evaluation metric.

x. Batch Size: 32, balancing computational efficiency and model convergence.

xi. Epochs: 100 with early stopping, providing sufficient iterations for learning.

Model Training and Testing

After preprocessing, the dataset was divided into two parts: the training set and the testing set, with a ratio of 75:25. The training set comprised 75% of the total dataset and was used to fit and build the model. The testing set served as an unbiased evaluation of the final model, which was trained on the training dataset. It provided the gold standard used to assess the performance of the model. The testing set was utilized only after the model had been completely trained. The combined architecture is shown in Figure 3.

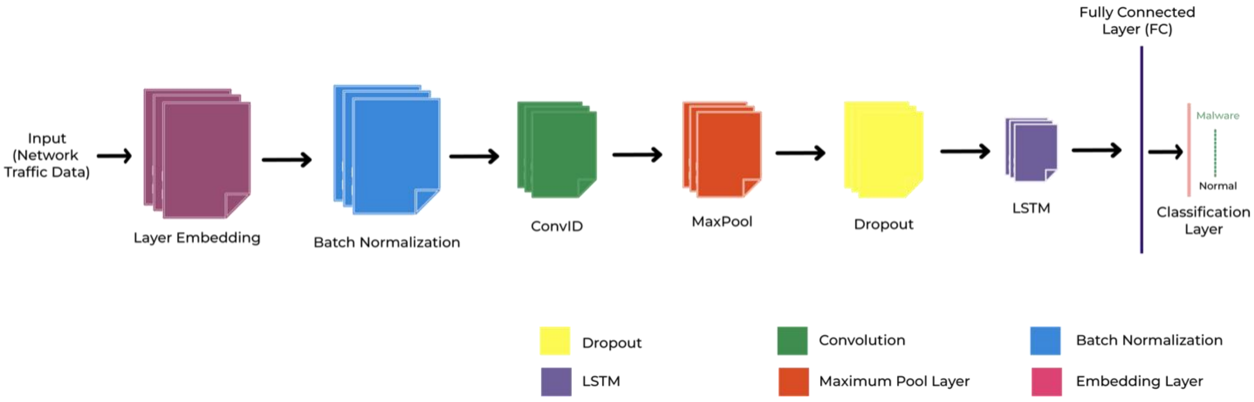

Figure 3: CNN-LSTM Combined Model Architecture

Evaluation Metrics

The experimental comparison of classification algorithms was done based on the performance measures of classification accuracy, precision, recall, and F1-score. The model was evaluated based on the following metrics:

Confusion Matrix

A classification matrix was generated and represented by a confusion matrix, displaying the actual and predicted classifications. A confusion matrix is a tabular representation utilized to characterize the efficacy of a classification model on a dataset with known true values.

Following the generation of the confusion matrix for each implemented algorithm, the subsequent metric values—Accuracy, Precision, Recall, and F1-Score—were computed from the confusion matrix utilizing the formulas provided below. Table 2 presents the confusion matrix for a classifier with two distinct classes (Strasak, 2017).

Table 2: Confusion Matrix for Two Classifiers [15].

	Predicted Positive	**Predicted Negative**
Actual Positive	**TP**	**FN**
Actual Negative	**FP**	**TN**

Based on the values of the confusion matrix, we can calculate the various evaluations like precision, recall, F1-score and accuracy.

1. **Accuracy**: the accuracy percentage is defined as the ratio of correctly classified instances to the total number of instances = $\frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$ ....(1)
2. **Precision:** Precision is the proportion of true positive predictions among the total retrieved instances = $\frac{\text{TP}}{\text{TP} + \text{FP}}$..(2)
3. **Recall:** It is the proportion of positively predicted instances relative to the total instances = $\frac{\text{TP}}{\text{TP} + \text{FN}}$…................................(3)
4. **F1-Score:** This represents the measure of predictive performance calculated from the precision and recall of the model = $\frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$..(4)

RESULTS AND DISCUSSION

The experiments were designed and conducted according to the methodology outlined in Figure 1, utilizing the embedded shallow CNN-LSTM architecture illustrated in Figure 3. To ensure a comprehensive evaluation, the experiments were divided into two distinct phases, each focusing on different

These metrics will provide us with insights into the performance of the trained model and its capability to successfully classify different instances of our dataset.

aspects of the model's capabilities and potential limitations.

First Experiment: Baseline Model without Class Balancing

In the initial phase of the experiment, the combination of CNN and LSTM was applied without incorporating any data class balancing techniques. This setup served as a baseline to evaluate the raw performance of the model when faced with imbalanced datasets, a common challenge in malware classification tasks. By omitting class balancing methods, the experiment aimed to assess how well the model could generalise and classify malware samples under real-world conditions where class distributions may be skewed.

Second Experiment: Incorporating Class Balancing Techniques with CNN

In the second phase, SMOTE was employed to address the issue of class imbalance. However, this experiment focused exclusively on the CNN component of the architecture, temporarily excluding the LSTM layer. The goal was to isolate the impact of class balancing techniques on CNN's ability to

DOI: 10.56892/bima.v9i1A.1259

classify malware effectively. This allowed for a clearer understanding of whether the improvements in performance were due to the balancing methods or other factors.

CNN-LSTM with Imbalanced Class Model

Table 3: CNN-LSTM with Imbalance model classification report

	Precision	Recall	F1-score	Support
Benign	0.89	0.65	0.75	283
Malware	0.99	1.00	0.99	10686
Accuracy			0.99	10969
macro avg	0.94	0.83	0.87	10969
weighted avg	0.99	0.99	0.99	10969

The classification report results from the shallow hybrid CNN-LSTM model, as shown in Table 3, applied to an imbalanced dataset, disclose significant performance metrics that underscore the model’s strengths and weaknesses in differentiating between benign and malware samples. The assessment metrics encompass precision, recall, F1-score, and support, offering a thorough perspective on the model’s efficacy.

Starting with the benign class, the model exhibits a precision of 0.89, signifying that roughly 89% of the instances categorized as benign are accurately recognized. The recall value for this class is 0.65, indicating that only 65% of the actual benign instances are accurately identified by the model. This inconsistency indicates that although the model exhibits considerable confidence in its predictions for benign samples, it fails to accurately identify all genuine benign cases, possibly misclassifying some as malware. The F1-score, which equilibrates precision and recall, is 0.75 for the benign class, indicating a moderate overall performance level.

Conversely, the model exhibits outstanding performance for the malware category, attaining a precision of 0.99 and a recall of 1.00. The elevated values suggest that nearly all instances categorized as malware are genuinely harmful, and almost all true malware instances are accurately recognized. The F1-score for the malware category is 0.99, highlighting the model's efficacy in identifying malicious samples. The dataset is significantly imbalanced, comprising 10,686 malware samples and merely 283 benign samples; thus, the model's capacity to attain high accuracy for the majority class is anticipated yet remains remarkable.

The model attains a remarkable accuracy of 0.99, indicating its robust performance throughout the entire dataset.

Nevertheless, when evaluating the macro average, which assigns equal importance to both classes irrespective of their distribution, the precision decreases to 0.94, the recall to 0.83, and the f1-score to 0.87. This suggests that the model’s performance is biased towards the majority class (malware) because of the imbalance, resulting in a diminished capacity to manage the minority class (benign).

The weighted average, reflecting class distribution, closely corresponds with the overall accuracy, demonstrating precision, recall, and F1-score all at 0.99.

This further underscores the predominance of the majority class in shaping the overall performance metrics. The high accuracy indicates the model's overall effectiveness, yet

DOI: 10.56892/bima.v9i1A.1259

the performance disparity between the two classes underscores the difficulties associated with imbalanced datasets.

In conclusion, the shallow hybrid CNN-LSTM model demonstrates exceptional efficacy in malware identification, especially within a dataset predominantly composed of malware instances. Nonetheless, its capacity to identify benign samples is less reliable, probably owing to the class imbalance. These findings highlight the necessity of rectifying class imbalance in subsequent efforts to enhance the model's accuracy regarding minority classes. Moreover, although the model shows considerable promise for practical applications in cybersecurity, additional improvements are required to guarantee efficient and effective classification across all categories.

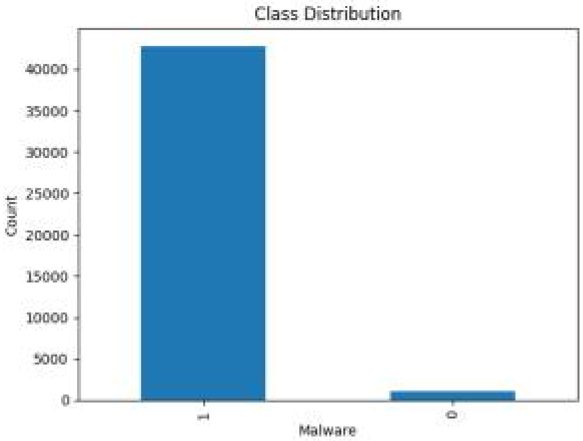

Figure 4: Dataset Class Distribution Before Balancing.

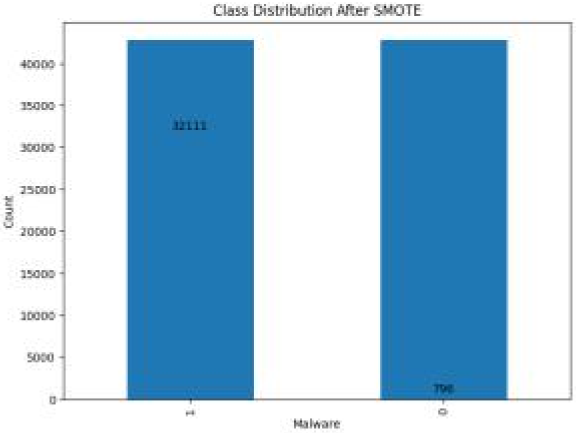

Figure 5: Dataset Class Distribution After Balancing.

Shallow Hybrid CNN – LSTM with Balanced Class Model

The dataset analysis revealed a significant class imbalance because the number of samples varied greatly between classes. To resolve this problem, SMOTE is adopted.

The minority class representation was expanded within the dataset by using SMOTE to create synthetic samples. The technique improved class distribution equity and minimized the model's majority class bias.

Figure 4 presents the original class distribution, which demonstrates the dataset's imbalance, while Figure 5 illustrates the class distribution after SMOTE was applied and demonstrates how it achieved a more balanced class representation.The figures demonstrate how SMOTE transformed the dataset by addressing class imbalance and creating an equitable dataset for analysis and successful model training.

Table 4: Hybrid Shallow CNN-LSTM Model Classification Report with Balanced classes

Classes	Precision	Recall	F1-score	Support
0	1	0.99	0.99	10722
1	0.99	1	0.99	10677
Accuracy				0.99
Macro Avg	0.99	0.99	0.99	21399
Weighted Avg	0.99	0.99	0.99	21399

DOI: 10.56892/bima.v9i1A.1259

The classification report obtained from the hybrid shallow CNN-LSTM model, evaluated on the balanced dataset with SMOTE shows outstanding performance across all principal metrics. The model was evaluated on a dataset with balanced representation of two classes (0 and 1), guaranteeing that the results accurately assess the model's ability to classify both majority and minority classes.

In class 0, denoting a specific category of instances (e.g., benign or non-malware), the model attains a precision of 1.00, signifying that every instance designated as class 0 is accurately recognized. A recall value of 0.99 indicates that the model accurately identifies 99% of all true instances of this class. The elevated F1-score of 0.99 further shows the model's robust performance between precision and recall for class 0.

For class 1 (e.g., malware), the model demonstrates a precision of 0.99, indicating that almost all instances categorized as class 1 are true positives. The model attains a recall of 1.00 accurately identifying all true instances of class 1, thereby achieving perfect sensitivity. The F1-score of 0.99 highlights the model's strong efficiency in managing this class.

The model's overall accuracy is reported at 0.99 indicating a high degree of correctness throughout the dataset. The remarkable accuracy is further supported by the macro average and weighted average metrics, which both attain scores of 0.99 for precision, recall, and F1-score. These averages offer a comprehensive perspective on the model's performance, affirming its consistent accuracy in classifying both classes with equal proficiency.

Comparison of Imbalanced Results with Balanced Data Results

Table 5 contains a summary of experimental results that evaluate CNN-LSTM model performance across both imbalanced and balanced datasets. The evaluation comparison highlights how class balancing methods affect critical performance metrics, including accuracy, precision, recall, and F1-score.

Table 5: Comparison of the Proposed Model with and without Balanced Dataset.

Models	Accuracy	Precision	Recall	F1-Score
CNN-LSTM (Imbalance)	99.22	0.94	0.83	0.87
CNN-LSTM (Balance)	99.44	0.99	0.99	0.99

The CNN-LSTM model assessed on the imbalanced dataset attains an accuracy of 99.22%, indicating robust overall performance. Nonetheless, its precision, recall, and F1-score indicate a significant imbalance in its capacity to manage the minority class. Precision is 0.94, recall is 0.83, yielding an F1-score of 0.87. The values suggest that the model has difficulty accurately identifying all instances of the minority class because of the inherent dataset imbalance.

The CNN-LSTM model assessed on the balanced dataset demonstrates markedly enhanced performance across all metrics. Balancing the dataset enables the classification model to achieve a high accuracy rate of 99.44%. Precision and recall reach almost perfect levels at 0.99 which produces an F1-score of 0.99.

DOI: 10.56892/bima.v9i1A.1259

Evaluation of the developed model's performance in comparison to existing models

Table 6 shows the comparison between the developed CNN-LSTM model and existing works within malware classification. The study presents an extensive assessment of the model's capabilities by comparing its results with those from previous researchers.

Table 6: Comparison of Proposed CNN-LSTM Model with Existing Models

Author	Method	Accuracy
Akarsh et al. (2019a)	CNN-LSTM	96.68%
Akarsh et al. (2019b)	1D CNN-LSTM	95.5%
Lu et al. (2020)	CNN-LSTM	94%
Di Troia et al. (2021)	LSTM and CNN	81%
Karat et al. (2024)	CNN-LSTM	96%
Proposed	CNN-LSTM with SMOTE CNN-LSTM	99.44% 99.22%

The study by (Di Troia et al., 2021) applied LSTM and CNN methods to reach 81% accuracy. This method shows deep learning techniques have potential for malware classification but performs worse than newer methods. Similarly, (Lu et al., 2020) achieved 94% accuracy using a combined CNN-LSTM model. The outcomes demonstrate substantial advancement in using hybrid models to improve classification accuracy.

Karat et al. (2024) developed a CNN-LSTM hybrid model for malware detection, achieving an accuracy of 96%. Furthermore, the model achieved 0.95 precision, a recall value of 0.95, and an F1-score of 0.95. This method shows deep learning techniques have potential for malware classification.

The state-of-the-art is pushed further by the developed CNN-LSTM model which includes class balancing techniques. The model reaches 99.44% accuracy when applied to balanced datasets, which demonstrates its effectiveness in managing both majority and minority classes. The model exhibits remarkable adaptability to different data distributions by sustaining a high accuracy rate of 99.22% even when tested against an imbalanced dataset.

The performance assessment shows that our developed model achieves better accuracy than existing methods. Class balancing techniques play a crucial role in enhancing performance, which makes this approach a viable solution for accurate malware detection. The findings demonstrate how this model can improve malware detection while effectively dealing with imbalanced datasets. The model represents an important breakthrough in cybersecurity because it surpasses earlier research achievements.

Summary

This work focuses on enhancing malware attack classification using a hybrid shallow convolutional neural network (CNN) embedded with long short-term memory (LSTM). The study addresses the challenge of class imbalance in malware datasets by comparing the performance of the CNN-LSTM model on both imbalanced and balanced datasets. A balancing technique i.e. SMOTE was employed to balance the dataset, significantly improving the model's ability to classify minority classes.

DOI: 10.56892/bima.v9i1A.1259

Experimental results demonstrate that the CNN-LSTM model achieves an accuracy of 99.44% on balanced data and 99.22% on imbalanced data, outperforming existing models in terms of precision, recall, and F1-score. This research highlights the importance of addressing class imbalance and showcases the potential of hybrid deep learning architectures for advancing malware detection and classification in cybersecurity.

CONCLUSION

This study successfully demonstrated the effectiveness of a hybrid shallow CNN-LSTM model in enhancing malware attack classification by addressing the challenge of class imbalance by adopting SMOTE. The model achieves superior performance compared to existing approaches. The results show that the CNN-LSTM model attains an accuracy of 99.44% on balanced datasets and maintains a high accuracy of 99.22% even on imbalanced datasets, highlighting its robustness and adaptability.

The comprehensive evaluation using metrics such as precision, recall, and F1-score, further validates the model's ability to accurately classify both majority and minority classes. These findings underscore the importance of balancing datasets in improving the reliability of machine learning models for cybersecurity applications.

While the proposed model demonstrates significant advancements in malware detection, there is still room for improvement, particularly in reducing false positives and false negatives. Future work could explore more advanced data augmentation techniques, feature engineering methods, or ensemble approaches to further enhance performance. Additionally, applying the model to larger and more diverse datasets could provide deeper insights into its scalability and generalisation capabilities.

REFERENCES

Abdullah, M. A., Yu, Y., Adu, K., Imrana, Y., Wang, X., & Cai, J. (2022). HCL-Classifier: CNN and LSTM based hybrid malware classifier for Internet of Things (IoT). Future Generation Computer Systems, 142, 41–58. https://doi.org/10.1016/j.future.2022.12.034

Akarsh, S., Poornachandran, P., Menon, V. K., & Soman, K. (2019). A detailed investigation and analysis of deep learning architectures and visualization techniques for malware family identification. In Cybersecurity and Secure Information Systems (pp. 241–286). Springer.

Akarsh, S., Simran, K., Poornachandran, P., Menon, V. K., & Soman, K. (2019). Deep learning framework and visualization for malware classification. Proceedings of the 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS) (pp. 1059–1063).

Akhtar, M. S., & Feng, T. (2022). Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time. Symmetry, 14(11), 2308.

Alzaylaee, M. K., Yerima, S. Y., & Sezer, S. (2020). DL-Droid: Deep learning-based android malware detection using real devices. Computers & Security, 89, 101663.

Awan, M. J., Masood, O. A., Mohammed, M. A., Yasin, A., Zain, A. M., Damaševičius, R., & Abdulkareem, K. H. (2021). Image-based malware classification using VGG19 network and spatial convolutional attention. Electronics, 10(2444). https://doi.org/10.3390/electronics10192444

Bensaoud, A., & Kalita, J. (2024). CNN-LSTM and transfer learning models for malware classification based on opcodes and API calls. Knowledge-Based Systems, 290, 111543.

Cui, Z., Du, L., Wang, P., Cai, X., & Zhang, W. (2019). Malicious code detection based on CNNs and multi-objective algorithm. Journal of Parallel and Distributed Computing, 129, 50–58.

Dang, D., Di Troia, F., & Stamp, M. (2021). Malware Classification Using Long Short-Term Memory Models. arXiv preprint. Retrieved from arXiv:2103.02746.

Hakak, S., Khan, W. Z., Imran, M., Choo, K.-K. R., & Shoaib, M. (2020). Have you been a victim of COVID-19-related cyber incidents? Survey, taxonomy, and mitigation strategies. IEEE Access, 8, 124134–124144.

Hindy, H., Atkinson, R., Tachtatzis, C., Colin, J.-N., Bayne, E., & Bellekens, X. (2020). Utilizing deep learning techniques for effective zero-day attack detection. Electronics, 9(1684).

Imamverdiyev, Y., & Baghirov, E. (2024). EVASION TECHNIQUES IN MALWARE DETECTION: CHALLENGES AND COUNTERMEASURES. Problems of InformationTechnology, 15(2), 9–15. https://doi.org/10.25045/jpit.v15.i2.02

Jain, Mugdha, "Image-Based Malware Classification with Convolutional Neural Networks and Extreme Learning Machines" (2019). Master's Projects. 900. https://doi.org/10.31979/etd.jand-r63

Karat, et al. (2024). CNN-LSTM hybrid model for enhanced malware analysis and detection. Elsevier.

Krishnamurthi, R., Kumar, A., Gill, S. S., & Xhafa, F. (2022). Autonomous and connected heavy vehicle technology. Elsevier.

Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-Based model to forecast stock prices. Complexity, 2020, 1–10. https://doi.org/10.1155/2020/6622927.