symmetry

MDPI

Article

Detection of Malware by Deep Learning as CNN-LSTM Machine Learning Techniques in Real Time

Muhammad Shoaib Akhtar and Tao Feng *

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China
* Correspondence: fengt@lut.edu.cn

Abstract: Cyber-attacks on the numerous parts of today's fast developing IoT are only going to increase in frequency and severity. A reliable method for detecting malicious attacks such as botnet in the IoT environment is critical for reducing security risks on IoT devices. Numerous existing methods exist for mining IoT networks for previously discovered patterns that may be exploited to improve security. This study used a hybrid deep learning approach, namely the CNN-LSTM technique, to detect botnet attacks. Any software that infiltrates a computer system or is installed there without the administrators' knowledge or permission is malicious. There is a wide range of viruses that cyber-criminals use to further their nefarious ends. A revolutionary deep learning system has been developed to counteract the increasing quantity of harmful programs. The system takes advantage of NLP methods as a baseline, mixes CNNs and LSTM neurons to capture local spatial correlations, and learns from successive long-term dependencies. Spatial invariance, often known as symmetry, is the property wherein the dataset size remains constant throughout iterations of an algorithm while undergoing various transformations. Therefore, automated extraction of high-level abstractions and representations aids in the malware categorization process. When compared to its predecessor research study, the current level of categorization accuracy is significantly greater than 0.81. The proposed CNN-LSTM method obtained an R2 = 99.19% in the dataset, with a correlation coefficient for the CNN-LSTM technique of R2 = 100% utilizing the provided dataset. The symmetry correlation of the CNN-LSTM, which illustrates that the CNN-LSTM method has the highest detection accuracy, at 99%, among the other malware detection methods such as the SVM and DT. The rest of classifiers had an accuracy of 98% for DT, and 95% for SVM. The accuracy of the LSTM model is 99%, the precision of the CNN-LSTM is 99%, recall is 99% and F1 score is 1.

Keywords: CNN-LSTM; cyber-attacks; malware; Iot; malicious threats; machine learning algorithms; cyber security; suspicious activity; cyber threats; malware detection

Citation: Akhtar, M.S.; Feng, T. Detection of Malware by Deep Learning as CNN-LSTM Machine Learning Techniques in Real Time. *Symmetry* **2022**, *14*, 2308. https://doi.org/10.3390/sym14112308

Academic Editor: José Carlos R. Alcantud

Received: 29 September 2022
Accepted: 28 October 2022
Published: 3 November 2022

1. Introduction

Cyberattacks from hackers are currently the leading cause of concern in the technological world. The word refers to the action of exploiting a security flaw for malicious purposes, such as gaining unauthorized access to data, changing a code, or destroying the system. Malicious software is one example of a cyberattack. Malware is any program or collection of instructions designed to damage a computer, user, business, or computer network [1]. Malicious software, or malware, is any program designed to cause harm, such as a computer virus, Trojan horse, ransomware, spyware, adware, rogue program, data eraser, scareware, and many more [2]. Dangerous software is any piece of code that runs on a computer without the user's knowledge or consent. Malware detection modules are designed to assess whether or not a specific piece of software or network connection constitutes a security risk based on the data they have collected and been educated on [3]. Take, as an example, a machine learning system that can explain the overarching principles it has observed. Machine learning algorithms may improve their prognostic skills by analyzing the data from their training sessions

A. Contribution

- The main focus of this research study is to use the CNN-LSTM method and detect the malware of IoT devices in the emerging technology [4].
- The proposed CNN-LSTM method aims to detect malware in real-time.
- The author divided the dataset in to two sets, one set is employed for the training the dataset which is composed of 70% of the data, the second set represents 30% of the data and is used to test the trained CNN-LSTM model.
- Our deep learning model achieves an accuracy rate of 99% without the use of any complicated feature engineering.

Worldwide, cybercriminals pose a threat to businesses, governments, and individuals by spreading malicious software and stealing private information [5]. Every day, hundreds of hackers employ malicious software to breach networks, steal data, and make unlawful financial transactions. Because of this, the security of personal information is becoming a pressing concern in the scientific world. This study uses data mining and machine learning classification methods to identify malicious software and prevent its access to sensitive information [6]. To accomplish this, we perform an analysis of signature-based and anomaly-based features in order to provide a trustworthy and effective way for malware classification and detection. The experimental results have proven that the proposed strategy is superior to the alternatives.

Multiple cyberattacks, such as those depicted in Figure 1, can be launched in the context of cyberwarfare. Today's malware is prevalent and complex; as a result, it presents a significant threat to the security of online platforms [7]. Malicious software, or malware, is software with the intent of harming a computer or network in some way, most commonly for financial gain. Malware attacks increasingly target IoT devices, medical gear, and infrastructure control systems in both natural and constructed settings [8]. Modern spyware is notoriously hard to detect since it constantly updates its code and behavior. The proliferation of malware has reached a stage where traditional signature-based defenses are ineffective. There needs to be a broader set of safeguards in place instead of just ignoring these cyber threats [9].

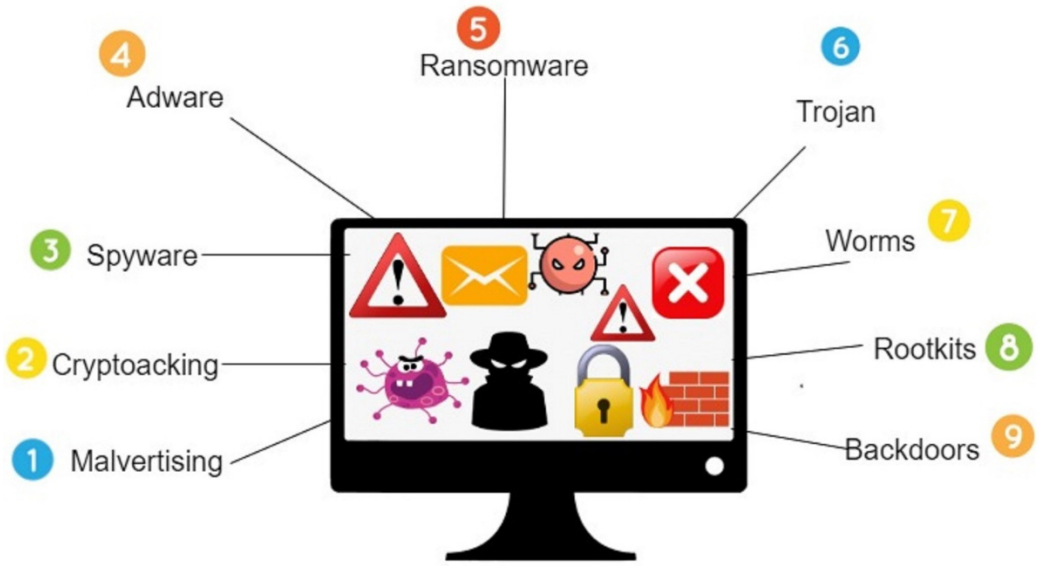

Figure 1. Different types of malware attacks.

The identification and classification of malware have become more difficult tasks in recent years [9]. Polymorphism and metamorphism are only two of the many strategies used by malware writers to confuse detectors and avoid "pattern matching" detection. Malware analysis might benefit from monitoring actual infection rates and pattern mining based on the virus changing behaviors in the field [10].

Malware attackers have developed a number of automated malware-generating toolkits, allowing for the quick manufacture of malicious programs with an infinite number of variations that may easily circumvent standard pattern matching detections. Since this is the case, we need cutting-edge methods to automatically analyze malware, find trends, and stop common ways of avoiding detection. In recent years, deep learning's popularity has surged as a tool for tackling difficult pattern recognition and machine learning problems [11]. With the help of these local representations of features, it is possible to learn and remember higher-level abstractions. In this paper, we compare the order of API calls to the syntax of a sentence. We proposed a new model for malware classification that makes use of deep neural networks and a model of natural language processing to improve classification accuracy.

The number of organizations (including corporations, banks, universities, social media accounts, and government departments) that rely on internet connectivity is increasing at an exponential rate. This [12] growth might be jeopardized by cybercriminals who use malicious software and network threats in their attacks. When triggered, malware instructs a computer to perform abnormal operations, which may compromise the victim's data or software. In recent years, malicious software (or "malware") has become increasingly pervasive and damaging to computer systems worldwide. Every day, thousands of new malicious programs are created. Figure 2 depicts annual data from malware attacks over the last decade, indicating that more than 900 million pieces of malware were in circulation in 2019, representing a nearly 2000% increase from 2010. Even the smallest businesses may lose millions of dollars due to malware infestations. The identification of viruses requires more than merely routing protocols, which is a major drawback. Therefore, machine learning is used by researchers and antivirus vendors to detect and classify infections. Many studies have focused on binaries as a subset of malware because of how frequently they are used to infect systems. Malware evaluation may be conducted both statically and dynamically. The malware's dynamic behavior is analyzed in a sandbox while static analysis is utilized to extract features of the virus that may be used for detection or classification by machine learning [13].

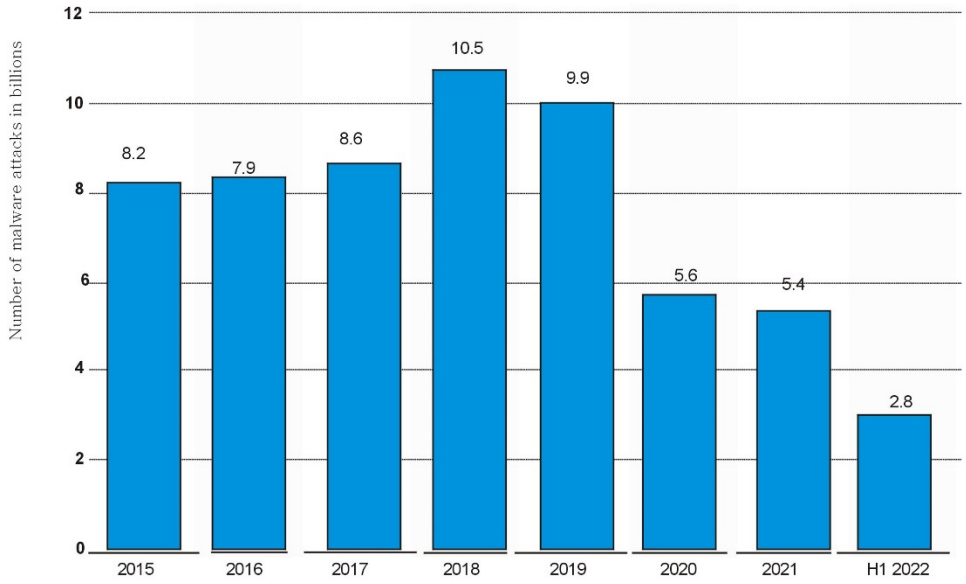

Figure 2. Worldwide malware cyber-attacks.

2. Literature Review

Deep learning is a subfield of machine learning that has garnered increasing attention from academic and corporate researchers in recent years. Originally developed for use in computer vision, deep learning has now expanded to other fields [14]. Voice recognition and NLP used as extra feature engineering methods are two areas where deep learning has

demonstrated very strong potential [15]. Deep learning is different from more traditional, shallow machine learning methods in that it can benefit from starting with the raw data itself, thereby doing away with the time-consuming task of feature generation by hand [16]. Deep neural network CNN-LSTMs also collect representations from different levels, which are abstractions of different layers, by stacking several layers in a hierarchical way. Tuning parameters across several layers allows for training a model with more granularity [17].

Several types of deep learning networks have been used in a wide range of settings, employing a broad variety of datasets [18]. Creating networks and unit operations in a number of different methods is required for capturing and learning features from a wide range of sources [19]. For example, CNNs (convolutional neural networks) excel in processing visual and aural data because of their capacity to operate in a two-dimensional plane. RNNs have excelled in the field of natural language processing. Even with a small amount of training data, RBM (restricted Boltzmann machines)-built generic DBN (deep belief networks) are good at modeling and fine-tuning convergence speed [20]. Malware is a major threat to cyber security on all scales, if not identified quickly after it has been introduced. Malware is proliferating at an alarming rate, making it difficult for even highly experienced network administrators to recognize it, much less typical internet users. Traditional detection approaches focused on feature extraction and comparison are becoming more useless as a result [21].

Chen's [22] research results depict that the accuracy of the Chen [22] proposed CNN method for malware detection is 91.01%. Malicious software comes in various forms, including source code, binary files, Perl scripts, shell scripts, instructions, and others, and its complexity has only grown over the years, making detection even more of a Herculean task. Because of the added intricacy, mistakes in judgement are punished more heavily.

The Chen [22] CNN method testing accuracy, which is 91.01% even when the FPR is 21%, remained same as in training of the CNN-LSTM method for malware detection. In this study, convolutional neural networks (CNNs), one of the most successful deep learning methodologies, are used. The experimental findings show a success rate of over 90% in differentiating malicious from safe programs. In addition to recognizing binary and source code, CNN was able to identify harmful code that had been introduced into otherwise safe code, as shown by the experiment. In today's tough network environment, IT workers need a way to take preventative steps and plan for future cyberattacks, so this study suggests a practical way to find malware at its source.

Luo proposed a CNN-LSTM method tested on the small dataset. It is clear from the results that by decreasing FPR to 0.2% the accuracy of the Chen [22] proposed LSTM-CNN method increased to 99.6%, but this accuracy was recorded when the dataset is completely normalized and the dataset is small.

3. Research Problem

Threat components of malware can be identified by static analysis or dynamic analysis. The goal of both static analysis and the reverse engineering technique used to decompile the virus is to parse the malware files and locate the malicious strings inside them. Dynamic analysis, on the other hand, involves keeping an eye on malicious code even as it runs in a secure setting such as a virtual machine. Although each approach has benefits and drawbacks, using both is recommended when analyzing malware. If malware detection were to improve, it would be because fewer harmful traits were used in its creation. We worry that a plethora of features are being utilized to identify malware when a smaller set of traits with greater reliability is sufficient. Finding potential techniques or algorithms is the first step in picking which harmful features to employ [23]. There needs to be a way to both greatly reduce the number of features needed to find malware and find malware that has never been seen before.

4. Research Framework

Deep learning requires no labeled information to identify its next move. The CNN-LSTM method reduces the need for time-consuming feature engineering, which is needed for shallow machine learning techniques but might not be able to pull out enough useful features for classification tasks, as well as relationships between variables including dependency, consistency, and structural information in the collected data [24]. When trained on the same data, CNNs and RNNs may pick up representations from different angles and capture different attributes at higher levels. Convolutional neural networks (CNNs) excel at capturing local spatial correlation, but recurrent neural networks (RNNs) are particularly strong at gathering temporally sequential data. The CNN's strength in this area lies in its ability to extract features at each position in the sequence while sliding over the entire series via convolution operations between filters and the sequence, making it ideal for modeling the API call sequence. Long short-term memory (LSTM) can automatically figure out long-term relationships from a set of random data points [25].

An overall representation of the suggested mixed deep model is depicted in Figure 3, combining both the temporal and geographical interactions of the CNN-LSTM, at the micro and macro levels. The CNN-LSTM model solves the problem of classifying malware by automatically abstracting and expressing high-level n-gram API requests as sequential feature maps.

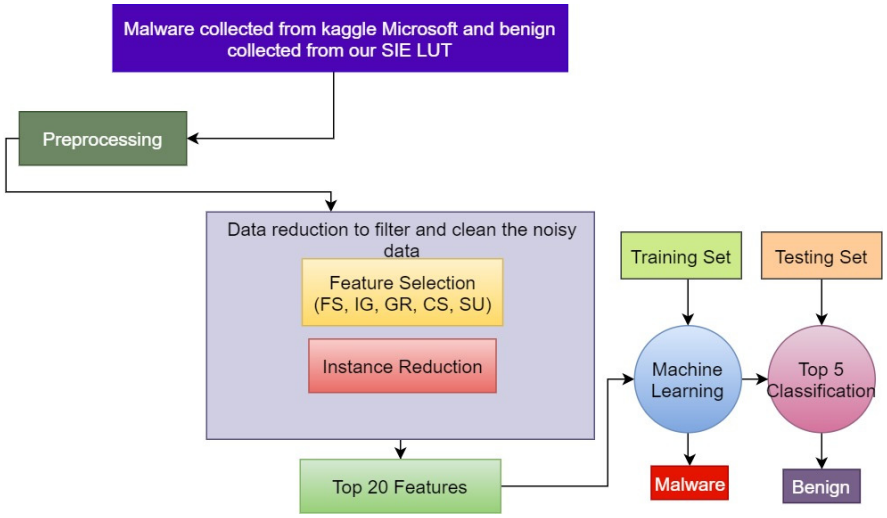

Figure 3. Proposed method of malware detection.

5. Research Methodology

Figure 4 depicts a high-level outline of our machine learning-based malware detection process. Classifier training, advanced malware detection, and feature selection from promising datasets all figure into this procedure. The methodology that was used in this investigation is described in further depth below [26].

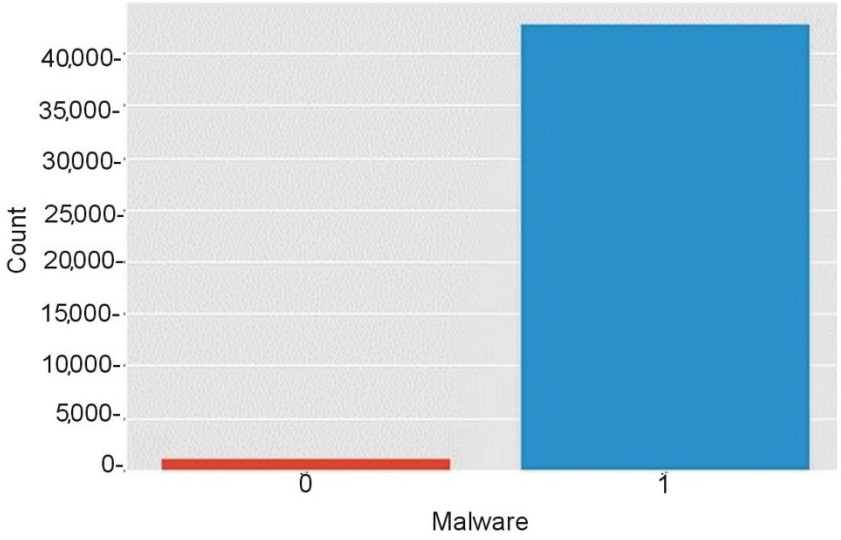

Figure 4. Malware after feature selection.

5.1. Dataset

The study relied entirely on data collected on the website Kaggle. Various pieces of malware have stolen log information, which is included in many of the files in the collection. Models may be trained using the recovered log features in a wide variety of ways [22]. It was discovered that five distinct families of malware were included in the samples. Over 43867 individual pieces of information acquired from diverse sources are included. There are 100 columns and 5 rows in the data set as shown in Figure 5 [27].

	hash	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	...	t_91	t_92	t_93	t_9
0	071e8c3f8922e186e57548cd4c703a5d	112	274	158	215	274	158	215	298	76	...	71	297	135	17
1	33f8e6d08a6aae939f25a8e0d63dd523	82	208	187	208	172	117	172	117	172	...	81	240	117	71
2	b68abd064e975e1c6d5f25e748663076	16	110	240	117	240	117	240	117	240	...	65	112	123	65
3	72049be7bd30ea61297ea624ae198067	82	208	187	208	172	117	172	117	172	...	208	302	208	30
4	c9b3700a77facf29172f32df6bc77f48	82	240	117	240	117	240	117	240	117	...	209	260	40	20

Figure 5. Dataset preview. Preview of dataset for malware detection, we are using google colaboratory compiler, and for data preview we use command data.head.

5.2. Features Extraction

For feature extraction from the dataset, we are using utilizing deep neural networks (the author used embedded CNN layers) for feature extraction which is illustrated in Figure 6. In the twenty-first century, it is not unusual for datasets to include tens of thousands of features [28]. As the number of features included in a machine learning model grows, the phenomenon of overfitting has come into sharper focus in recent years. As a workaround, we condense the original, larger number of qualities into a more manageable set by selecting just the most pertinent details to provide. The goal of this study is to improve the existing dataset by picking out the most important static and dynamic features as shown in Figure 6 and getting rid of the rest [29].

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	...	t_90	t_91	t_92	t_93	t_94	t_95	t_96	t_97	t_98
0	112	274	158	215	274	158	215	298	76	208	...	117	71	297	135	171	215	35	208	56
1	82	208	187	208	172	117	172	117	172	117	...	60	81	240	117	71	297	135	171	215
2	16	110	240	117	240	117	240	117	240	117	...	123	65	112	123	65	112	123	65	113
3	82	208	187	208	172	117	172	117	172	117	...	215	208	302	208	302	187	208	302	228
4	82	240	117	240	117	240	117	240	117	172	...	40	209	260	40	209	260	141	260	141

Figure 6. Feature extraction from dataset.

5.3. Features Selection

Feature selection follows feature extraction, which entails the discovery of even additional features [30]. Choosing features from a pool of newly recognized attributes is called feature selection, and it plays a significant role in improving accuracy, simplifying the model, and reducing overfitting. In the past, researchers have attempted to spot malicious software by employing a wide variety of feature categorization algorithms. In this research, the feature rank approach is used a lot because it works well to choose relevant features for building malware detection models [31].

6. Results and Discussion

The training and testing phases constitute the backbone of any classification method. In order to train the system, it must be exposed to both harmful and safe data [31]. A classifier may be trained using machine learning techniques to automatically generate reliable predictions. With each new set of labeled data, the classifier, the CNN-LSTM model, improves its performance. The classifier is given a collection of new files, some of which are malicious and some of which are not, and is asked to assign a category to each one during the validation phase as shown in Figure 7 [32].

```
model = Sequential(name="Cnn-Lstm_model")
model.add(Embedding(input_dim=unique_api_calls, output_dim=8,
                    input_length=X_train.shape[1], name='layer_embedding'))
model.add(BatchNormalization())
model.add(Conv1D(filters = 32, kernel_size = 9, padding = 'same', activation = 'relu'))
model.add(MaxPool1D(pool_size = 2))
model.add(LSTM(units=512, return_sequences=False, dropout=0.2))
model.add(Dense(units=1, activation='sigmoid'))
```

Figure 7. CNN-LSTM model. [23].

Figure 7 illustrate the CNN-LSTM model. The CNN-LSTM model employs dropout at the last fully connected layer. Dropout does not seem to be a method for regularization as much as it seems to be a way to add layers to the whole model as illustrated in Figure 8 [33].

This section contains the experimental findings from our assessment of the effectiveness of the malware categorization and detection method we suggested [34]. The generated malware and cleanware datasets are used in experiments as shown in in Figure 8. In this work, we look at malware and put it into categories by using supervised machine learning with the CNN-LSTM classifier [35].The results see Figure 9.

```
Model: "Cnn-Lstm_model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
layer_embedding (Embedding)  (None, 100, 8)            2456
_________________________________________________________________
batch_normalization (BatchNo (None, 100, 8)            32
_________________________________________________________________
conv1d (Conv1D)              (None, 100, 32)           2336
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 50, 32)            0
_________________________________________________________________
lstm (LSTM)                  (None, 512)               1116160
_________________________________________________________________
dense (Dense)                (None, 1)                 513
=================================================================
Total params: 1,121,497
Trainable params: 1,121,481
Non-trainable params: 16
_________________________________________________________________
```

Figure 8. CNN-LSTM model summary.

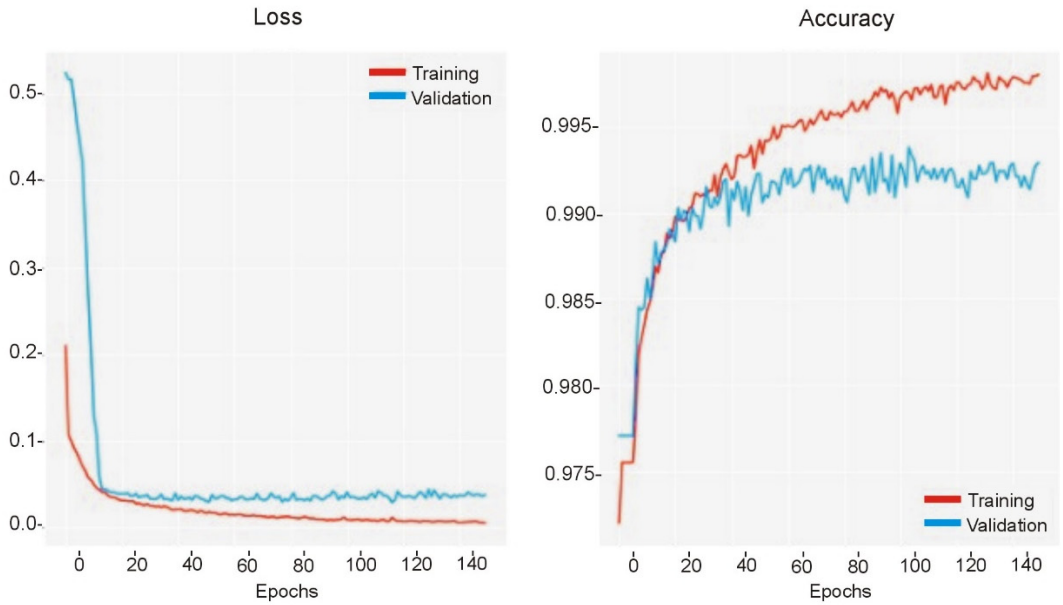

Figure 9. CNN-LSTM classifier results.

Figure 10 illustrates two trending lines: the red line represents the training of the dataset and the blue line represents the validation of the dataset [28]. After balancing the dataset, the precision of the training of the proposed model is 0.95, recall 0.87, and the average weighted accuracy is 99% [36].

	precision	recall	f1-score	support
0	0.90	0.74	0.81	283
1	0.99	1.00	1.00	10686
accuracy			0.99	10969
macro avg	0.95	0.87	0.90	10969
weighted avg	0.99	0.99	0.99	10969

Figure 10. CNN-LSTM model report.

Figure 11 shows the accuracy, recall, and F1 score of the proposed CNN-LSTM model. This study highlights the increasing attention that academics are paying to ML algorithmic approaches for malware detection [37]. Here, we offer a safeguard that uses three distinct machine learning (ML) algorithms to determine which is the most effective in detecting malware. According to the findings, the CNN-LSTM outperforms other classifiers in terms of detection accuracy, with 99% accuracy, 99% precision, 99% recall, and an F1 score of 1 [38].

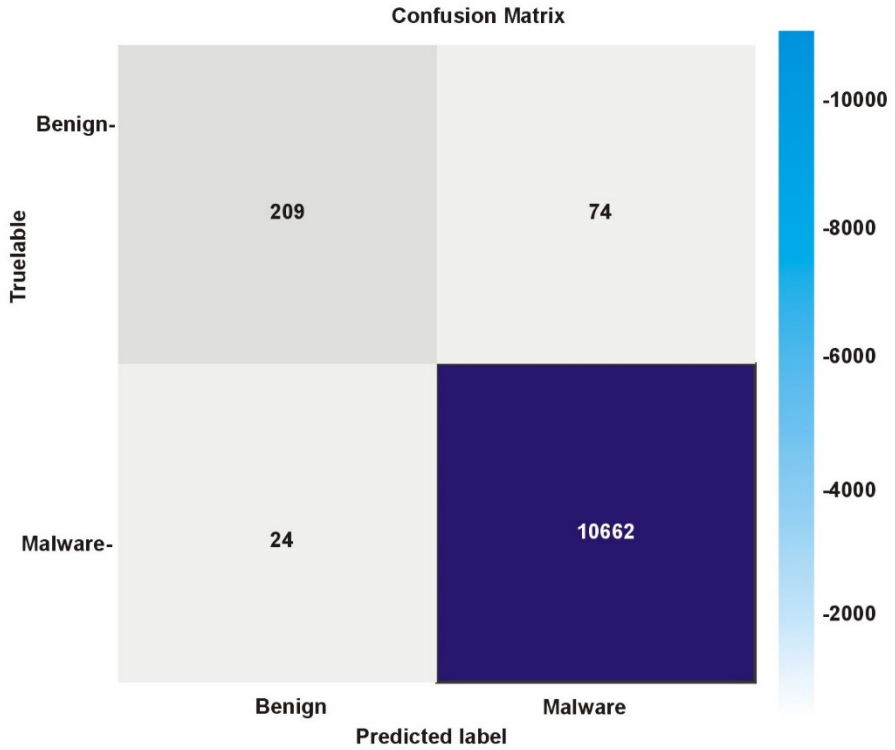

Figure 11. Illustration of the CNN-LSTM malware detection confusion matrix.

Figure 11 helps us visualize the confusion matrix of the CNN-LSTM proposed model. The prevalence and sophistication of malicious software are both on the rise. In this experiment, we compare an ML classifier trained on Python language on Google Colab python compiler [38,39]. The data sets are analyzed using static analysis against another ML classifier trained on other data sets in order to evaluate and quantify the detection accuracy of each [40]. Thanks to our research, machine learning algorithms can now identify harmful data from safe data. The CNN-LSTM machine learning approach's 99% accuracy was the best of any classifier we tested. The results of the experiments show that proposed CNN-LSTM model for malware detection accuracy is 99% when tested on python language [41].

Table 1 illustrates that the CNN-LSTM model has the highest detection accuracy (99%) among the other malware detection methods. DT accuracy is 98% and SVM accuracy is 95%.

Table 1. CNN-LSTM method comparison.

Method	TPR	FPR	Detection Accuracy
CNN-LSTM	1	0.0031	99%
DT	0.99	0.0039	98%
SVM	0.97	0.0043	95%

Limitation

Raw binary files' semantics are not taken in to account. The spatial patterns of each class of malware are in the raw binary files, and our tests show that deep learning models can use these patterns to correctly identify the class of a malware file.

7. Conclusions

In this research study, through the utilization of the CNN-LSTM to overcome major malware detection deficiencies, including the inefficiency of human feature building and the limitations of existing learning algorithms, we built a novel deep neural network ensemble by stacking CNN and LSTM techniques. The proposed CNN-LSTM method is used for the detection of advance malware without any feature engineering. Table 1, provided above, illustrates that the CNN-LSTM has the highest detection accuracy, which is 99%, among the other malware detection methods. For the rest of the classifiers, accuracy is 98% for DT and 95% for SVM. The accuracy of the LSTM model is 99%, precision of the CNN-LSTM is 99%, recall accuracy is 99%, and F1 score is 1. With the proposed CNN-LSTM model, malware detection accuracy is improved during training to around 1, with testing accuracy extremely close to training accuracy. All of this is possible thanks the interplay between the skills of a CNN-LSTM to extract spatially local correlations and the ability of a CNN-LSTM to represent sequences and learn from long-term dependencies.

Author Contributions: M.S.A. and T.F. contributed equally to the study's conception. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (No. 62162039) and (61762060), and the Key Research and Development Program of Gansu Province (No. 20YF3GA016).

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this research study.

CNN	Convolutional Neural network
LSTM	Long short-term memory
IoT	Internet of things
DBN	Deep Belief Networks
RNN	Recurrent Neural Network
FPR	False Positive Rate
RBM	Restricted Boltzmann Machine
DT	Decision Tree
SVM	Support Vector Machine

References

1. Nasiri, E.; Berahmand, K.; Li, Y. Robust graph regularization nonnegative matrix factorization for link prediction in attributed networks. *Multimedia Tools Appl.* **2022**. [CrossRef]
2. Li, D.; Li, Q.; Ye, Y.; Xu, S. Enhancing robustness of deep neural networks against adversarial malware samples: Principles, framework, and aics'2019 challenge. *arXiv* **2018**, arXiv:1812.08108.
3. Berahmand, K.; Nasiri, E.; Forouzandeh, S.; Li, Y. A preference random walk algorithm for link prediction through mutual influence nodes in complex networks. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 5375–5387. [CrossRef]
4. Zamiri, M.; Bahraini, T.; Yazdi, H.S. MVDF-RSC: Multi-view data fusion via robust spectral clustering for geo-tagged image tagging. *Expert Syst. Appl.* **2021**, *173*, 114657. [CrossRef]
5. Aleesa, A.M.; Zaidan, B.B.; Zaidan, A.A.; Sahar, N.M. Review of Intrusion Detection Systems Based on Deep Learning Techniques: Coherent Taxonomy, Challenges, Motivations, Recommendations, Substantial Analysis and Future Directions. *Neural Comput. Appl.* **2019**, *32*, 9827–9858. [CrossRef]
6. Apruzzese, G.; Colajanni, M.; Ferretti, L.; Guido, A.; Marchetti, M. On the effectiveness of machine and deep learning for cyber security. In Proceedings of the 2018 10th International Conference on Cyber Conflict (CyCon), Tallinn, Estonia, 29 May–1 June 2018; pp. 371–390.
7. Ferrag, M.A.; Maglaras, L.; Moschoyiannis, S.; Janicke, H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *J. Inf. Secur. Appl.* **2020**, *50*, 102419. [CrossRef]
8. Wickramasinghe, C.S.; Marino, D.L.; Amarasinghe, K.; Manic, M. Generalization of deep learning for cyber-physical system security: A survey. In Proceedings of the IECON 2018—44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 745–751.
9. Xu, X.; Liu, Q.; Zhang, X.; Zhang, J.; Qi, L.; Dou, W. A blockchain-powered crowdsourcing method with privacy preservation in mobile environment. *IEEE Trans. Comput. Soc. Syst.* **2019**, *6*, 1407–1419. [CrossRef]
10. Vinayakumar, R.; Soman, K.P.; Poornachandran, P. Evaluating effectiveness of shallow and deep networks to intrusion detection system. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 1282–1289.
11. Shone, N.; Ngoc, T.N.; Phai, V.D.; Shi, Q. A deep learning approach to network intrusion detection. *IEEE Trans. Emerg. Top. Comput. Intell.* **2018**, *2*, 41–50. [CrossRef]
12. Feng, Z.; Shuo, C.; Wang, X. Classification for DGA-based malicious domain names with deep learning architectures. *Int. J. Intell. Inf. Syst.* **2017**, *6*, 67–71.
13. Sahay, S.K.; Sharma, A. Grouping the Executables to Detect Malwares with High Accuracy. *Procedia Comput. Sci.* **2016**, *78*, 667–674. [CrossRef]
14. Kaggle. Microsoft Malware Classification Challenge (BIG 2015)" Microsoft. Available online: https://www.kaggle.com/c/malwareclassification (accessed on 10 December 2016).
15. Naseer, S.; Saleem, Y.; Khalid, S.; Bashir, M.K.; Han, J.; Iqbal, M.M.; Han, K. Enhanced network anomaly detection based on deep neural networks. *IEEE Access* **2018**, *6*, 48231–48246. [CrossRef]
16. Aftergood, S. Cybersecurity: The cold war online. *Nature* **2017**, *547*, 30–31. [CrossRef]
17. Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.K.; Du, X.; Ali, I.; Guizani, M. A survey of machine and deep learning methods for internet of things (IoT) security. *IEEE Commun. Surv. Tutor.* **2018**, *22*, 1646–1658. [CrossRef]
18. Feng, T.; Muhammad Shoaib, A.; Zhang, J. The future of Artificial Intelligence in Cybersecurity: A Comprehensive Survey. *EAI Endorsed Trans. Creative Technol.* **2021**, *8*, e3. [CrossRef]
19. Binbusayyis, A.; Vaiyapuri, T. Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM. *Appl. Intell.* **2021**, *51*, 7094–7108. [CrossRef]
20. Berman, D.S.; Buczak, A.L.; Chavis, J.S.; Corbett, C.L. A survey of deep learning methods for cyber security. *Information* **2019**, *10*, 122. [CrossRef]
21. Zhang, M.; Xu, B.; Bai, S.; Lu, S.; Lin, Z. A deep learning method to detect web attacks using a specially designed CNN. In Proceedings of the 24th International Conference on Neural Information Processing, Guangzhou, China, 14–18 November 2017; pp. 828–836.
22. Wang, W.; Zhu, M.; Zeng, X.; Ye, X.; Sheng, Y. Malware traffic classification using convolutional neural network for representation learning. In Proceedings of the 2017 International Conference on Information Networking, ICOIN, Da Nang, Vietnam, 11–13 January 2017.
23. Yang, H.; Wang, F. Wireless network intrusion detection based on improved convolutional neural network. *IEEE Access* **2019**, *7*, 64366–64374. [CrossRef]
24. Pascanu, R.; Stokes, J.W.; Sanossian, H.; Marinescu, M.; Thomas, A. Malware classification with recurrent networks. In Proceedings of the 2015 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015.
25. Tang, D.; Tang, L.; Shi, W.; Zhan, S.; Yang, Q. Mf-cnn: A New Approach for Ldos Attack Detection Based on MultiFeature Fusion and Cnn. *Mob. Netw. Appl.* **2020**, *26*, 1705–1722. [CrossRef]
26. Staudemeyer, R.C. Applying long short-term memory recurrent neural networks to intrusion detection. *S. Afr. Comput. J.* **2015**, *56*, 136–154. [CrossRef]

27. Aygun, R.C.; Yavuz, A.G. Network anomaly detection with stochastically improved autoencoder based models. In Proceedings of the 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), New York, NY, USA, 26–28 June 2017; pp. 193–198.
28. Gharib, M.; Mohammadi, B.; Dastgerdi, S.H.; Sabokrou, M. Autoids: Auto-encoder based method for intrusion detection system. *arXiv* **2019**, arXiv:1911.03306.
29. Baychev, Y.; Bilge, L. Spearphishing malware: Do we really know the unknown? In Proceedings of the Detection of Intrusions and Malware, and Vulnerability Assessment 15th International Conference, DIMVA 2018, Saclay, France, 28–29 June 2018; pp. 46–66.
30. Akhtar, M.S.; Feng, T. A Systemic Security and Privacy Review: Attacks and Prevention Mechanisms over IOT Layers. *ICST Trans. Secur. Saf.* **2022**, *8*, e5. [CrossRef]
31. Zhang, Y.; Chen, X.; Jin, L.; Wang, X.; Guo, D. Network Intrusion Detection: Based on Deep Hierarchical Network and Original Flow Data. *IEEE Access* **2019**, *7*, 37004–37016. [CrossRef]
32. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6. [CrossRef]
33. Jung, J.; Kim, H.; Shin, D.; Lee, M.; Lee, H.; Cho, S.-J.; Suh, K. Android Malware Detection Based on Useful API Calls and Machine Learning. In Proceedings of the 2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), Laguna Hills, CA, USA, 26–28 September 2018; pp. 175–178. [CrossRef]
34. Rana, S.; Gudla, C.; Sung, A.H. Evaluating Machine Learning Models for Android Malware Detection: A Comparison Study. In Proceedings of the 2018 VII International Conference on Network, Communication and Computing (ICNCC 2018), Taipei City, Taiwan, 14–16 December 2018; Association for Computing Machinery: New York, NY, USA; pp. 17–21. [CrossRef]
35. Rehman, Z.-U.; Khan, S.N.; Muhammad, K.; Lee, J.W.; Lv, Z.; Baik, S.W.; Shah, P.A.; Awan, K.; Mehmood, I. Machine learning-assisted signature and heuristic-based detection of malwares in Android devices. *Comput. Electr. Eng.* **2018**, *69*, 828–841. [CrossRef]
36. Rieck, K.; Holz, T.; Willems, C.; Düssel, P.; Laskov, P. Learning and classification of malware behavior. In Proceedings of the DIMVA 2008: Detection of Intrusions and Malware, and Vulnerability Assessment, 5th International Conference, DIMVA 2008, Paris, France, 10–11 July 2008; Springer: Berlin/Heidelberg, Germany, 2018; pp. 108–125.
37. Chen, C.-M.; Wang, S.-H.; Wen, D.-W.; Lai, G.-H.; Sun, M.-K. Applying Convolutional Neural Network for Malware Detection. In Proceedings of the 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), Morioka, Japan, 23–25 October 2019; pp. 1–5. [CrossRef]
38. Akhtar, M.S.; Feng, T. Deep Learning-Based Framework for the Detection of Cyberattack Using Feature Engineering. *Secur. Commun. Netw.* **2021**, *2021*, 6129210. [CrossRef]
39. Akhtar, M.S.; Feng, T. Comparison of Classification Model for the Detection of Cyber-attack using Ensemble Learning Models. *EAI Endorsed Scal. Inf. Syst.* **2022**, *22*, e6. [CrossRef]
40. Luo, S. Android Malware Analysis and Detection Based on Attention-CNN-LSTM. *J. Comput.* **2019**, 31–43. [CrossRef]
41. Coleman, S.-P.W.; Hwang, Y.-S. Malware Detection by Merging 1D CNN and Bi-directional LSTM Utilizing Sequential Data. In *Information Science and Applications*; Springer: Berlin/Heidelberg, Germany, 2021. [CrossRef]