International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

An Explainable Machine Learning Model for Predicting Unknown Malware Using Network Traffic

Hussein Ali Ghadhban Salman[1], Amir Jalaly Bidgoly[2], And Somayyeh Fallah.[3]

Submitted: 12/05/2024 **Revised**: 25/06/2024 **Accepted**: 05/07/2024

Abstract: Due to the increasing complexity of malware threats, early detection of malware is a prominent issue in the world of network security. Identifying malware using artificial intelligence approaches has a huge perspective for the security of the cyber world. Despite the many researches that have been done in this field to detect malware, the lack of interpretability of artificial intelligence models leads to the fact that users do not have enough confidence in the results predicted by the model. Explaining the prediction of an AI model can be the right basis for judging the final result of AI methods. In this work, we propose a new explainable AI method to interpret and investigate which features can be more effective in detecting future emerging malware. For this purpose, we carry out checks in two stages. First, we examine the ability of deep and shallow models to correctly detect malware. The evaluation results showed that BLSTM and BLSTM-GRU deep learning models were able to detect unknown traffic attacks with a high accuracy of 89%. In the second step, we extract the effective features in the model that had the best performance. Then, we will interpret and explain why these features led to high accuracy in the output results.

Index Terms *Malware detection, explainable AI, machine learning, deep learning, unknown malware.*

I. INTRODUCTION

SMARTPHONES are regularly attacked by new mal- ware due to their popularity and ease of use. These attacks are designed in such a way that they use different techniques to bypass the current detection systems. Dynamic execution, code obfuscation, repackaging and encryption are examples of evasion techniques [1]. If there is no response to security threats at the right time, irreparable damage will be done. These damages can include data destruction, theft of personal and financial information, embezzlement, fraud, disruption of business processes, etc. Cybercrime is predicted to cost the world $8 trillion USD in 2023, according to Cybersecurity Ventures. If measured as a country, then cybercrime would be the third largest economy in the world after the United States and China [2]. In recent years, researchers have proposed various methods to detect attacks. Artificial intelligence-based malware detection has received more attention than other methods and has been widely used to combat cyberattacks [3], [4]. Explaining the prediction of an artificial intelligence model, which is a countermeasure to cyber threats, is of great importance. Especially in situations where the lack of correct prediction leads to large financial and even human losses. Malware detection methods based on AI have had a significant performance, however, explanations are rarely given about the prediction of the model, which causes insufficient confidence in the prediction results. Explainability about the results of a model makes cyber security experts gain more knowledge about attacks and this is important in identifying attacks and types of malware. In other words, AI-based methods are incapable of justifying the results (ranging from detection and prediction to reasoning and decision-making) and making them understandable to humans [5]. As a result, explainable artificial intelligence (XAI)

[1]*Department of Information Technology and Computer Engineering, University of Qom, Iran (e-mail: husseina@basrahaoe.iq)*
[2]*Department of Information Technology and Computer Engineering, University of Qom, Iran (e-mail: Jalaly@qom.ac.ir)*
[3]*Department of Information Technology and Computer Engineering, University of Qom, Iran (e-mail: S.Fallah@stu.qom.ac.ir)*
Corresponding author: A. Jalaly Bidgoly (e-mail: Jalaly@qom.ac.ir).

k

has emerged as an important topic that can be explained or interpreted to human users. In this paper, we present a new explainable artificial intelligence method that is able to investigate the reasons behind the predictions of malware detection models. The main goal in this research is to analyze and investigate the features that have a great impact on predicting the target class. This work aims to provide detailed answers to the following questions:

1) which features are robust enough to detect emerging malware in the future?
2) Why do machine learning models work so well on some features?
3) Can changing and manipulating robust features cause the malicious traffic detection system to fail?

To answer these questions, we use a suitable and efficient explainable machine learning model to extract explanations from the model's performance. In the proposed approach, a series of shallow and deep machine learning methods including decision tree, random forest, linear regression, Naive Bayes, GRU (Gated Recurrent Unit), LSTM (Long Short Term Memory), BLSTM (Bidirectional LSTM) and the combination of BLSTM-GRU are used to check the performance of each model in detecting unknown malicious traffic. Hence, a complete set of all possible statistical features is extracted from the traffic flows and then given as an input sequence to the machine learning models. Then, the robust features that have the most impact on the detection of unknown malicious traffic are identified. In the last step, we analyze and examine each of the effective features and check why these features were robust and had higher coefficients in the model's performance. At this step, artificial intelligence explanation methods including SHAP method are used. The remainder of the paper is organized as follows: Section 2 reviews related work. Section 3 details our study design. Section 4 presents case studies and experiments, and finally, the paper concludes in Section 5 with conclusions and future work.

II. LITERATURE REVIEW

In this section, firstly, the works that are based on artificial intelligence methods in the field of malware detection are reviewed, Secondly, researches based on explainable artificial intelligence methods are introduced.

A. AI-BASED MALWARE DETECTION METHODS

Much research has been done in the field of malware detection with artificial intelligence methods [6], some of which achieved an accuracy of over 99% [7]. Kim and his colleagues [8], using image-level data and code-level data, they presented a hybrid deep generative model for malware detection. The features extracted from both data are entered into the model and the accuracy has reached 97.47%. In another work by Xing and his colleagues [9], they presented a convolutional neural network model based on the structure of an automatic encoder for detecting and classifying malware. In this work, the byte code of various command methods was used statically, to be converted into image data in gray scale to display the features of malware. The performance of the proposed model in this research was 96% accurate. In [10] using analysis of running application APIs, they demonstrated meaningful relationships between API sequences. In this way, the intrinsic features of API sequences are extracted by representation learning, which can determine whether any software is malicious or not. Then Bi-LSTM module was used to extract the relationship information between APIs. The proposed model achieved 97.31% accuracy in this work. In another related work by Asam and colleagues [11], they proposed a new framework for detecting Internet of Things Malware (iMDA) based on CNN in IoT devices. The proposed framework is applied to a new benchmark dataset for IoT malware analysis using squeezing and boosting dilated CNN. The proposed architecture differentiates malware from benign based on texture, contrast and pattern changes and obtained the best result with 97.33% accuracy. In [12] a malware detection system using Bayesian probability method was introduced. In this system, permission-based features were extracted using static analysis. Then, in order to better detect malware, these features are used using information gain and chi-square algorithms were optimized. In this work, chi-square algorithm with 15 features created the best performance among the features with 91% accuracy. In another paper [13], a hybrid model of deep learning based on bidirectional-gated recurrent convolutional neural network unit (Bi-GRU-CNN) was proposed to detect IoT malware and classify IoT malware families. In the proposed model, byte sequences from an ELF binary program entry point were used as a feature to effectively identify and classify Internet of Things malware. The performance of the proposed model showed 100% accuracy for Internet of Things malware detection and 98% accuracy for malware family classification. In [14] proposes a combination model of deep learning and machine learning for malware detection. In the proposed model, first, benign and malicious portable executable (PE) files are represented as color images, and an image-based data set is generated. Then, using deep learning model, it extracts deep features from the color image. Finally, the malware is detected using support vector machine (SVM). The proposed approach achieved an accuracy of 99.06%. In another related work [15], malicious and benign files were represented as images. Then, a deep generative adversarial neural network (DGAN) was used to generate new malware from the training malware dataset. Malware images produced with original malware and images of benign files are pre-processed and important features of malware are extracted using

deep CNN model. The evaluation results showed that the proposed model achieved 99.8% accuracy. In [16] a model for detecting Android malware using machine learning algorithms is presented. The proposed model includes new static features such as permissions, application components, method tags, targets, packages, API calls, and services/receivers. Then they used six machine learning classifiers to improve the performance of the model. The results obtained from the proposed model showed an accuracy of 96.24%. Some work to detect malware using dynamic methods. In [17] power consumption and traffic data were used to detect malware. Ten supervised machine learning algorithms were used to classify malicious data from benign, and the random forest algorithm had the best performance compared to other algorithms, and the features related to power consumption also performed better than the features based on network traffic. In another related work [18], a multi-view learning approach called DeepCatra was proposed for Android malware detection. In this work, a bidirectional LSTM and a graph neural network were used as subnets, as well as features extracted from statically computed call traces leading to critical APIs resulting from public vulnerabilities. The output of the proposed model showed that the performance of the hybrid model was better than several advanced detection methods using CNN, LSTM and GCN.

B. EXPLAINABLE AI-BASED MALWARE DETECTION METHODS

In [5], [19], [20], an extensive study of the application of explainable artificial intelligence in the field of cyber security, including various attacks and challenges faced, has been conducted. In another work [21], they proposed a malware detection system that combined textual and visual features of network traffic, which helped the model perform better. To validate the proposed method, an explainable artificial intelligence (AI) experiment was conducted. Zhang and colleagues [22] conducted a comprehensive review of the state-of-the-art research in cybersecurity with XAI methods. In this work, they presented a summary and overview of the classification of XAI defensive applications against cyber-attacks. They also identified the challenges associated with XAI. In another work by Liu [23], it investigates the effect of time inconsistency on Android malware detection. This means that benign and malicious programs are randomly selected regardless of their time period (malware samples are selected from 2010, while benign samples are selected from 2020). The results of their investigation showed that time discrepancies between benign and malicious samples significantly increase the performance of the malware detection system. The author in [24], they proposed a deep learning-based ransomware detection method explainable using dynamic analysis that combines different sequences based on dynamic analysis. These sequences are given as input to the two-layer CNN model. Then they used two XAI models as LIME and SHAP. The results of the proposed model showed up to 99.4% true positive rate (TPR). Further investigations using explainable AI approaches were used to understand why ML-based malware detection approaches perform so well under time-inconsistency conditions. By reviewing other works that have been done in the field of explainable artificial intelligence, it was concluded that most of the research in this field had a general survey about the performance of XAI approaches in cyber security. None of these works have investigated comprehensive aspects of performance and different objectives in explainable artificial intelligence in malware detection. For example, in no work have they interpreted and explained which features made the model work well, and investigated how the feature works in detecting unknown malware.

III. METHODOLOGY

The purpose of this article is to conduct a detailed analysis and research on the results of the unknown malware detection model. with this analysis, a better understanding of how the model works can be obtained. In this section, firstly, an overview of the study plan is described with the help of explainable artificial intelligence techniques, then explanations about the dataset and features are provided.

A. PROPOSED MODEL OVERVIEW

The proposed model is based on network traffic analysis. This method models network traffic flow as a sequence of statistical features. The overview of the proposed model is shown in Figure 1. In ML-based malware detection method, it should be checked whether a network flow is classified as benign or malware. For this purpose, samples from two separate classes including benign and malware are needed. In this work, a complete set of statistical features of network traffic is extracted and fed to learning models. Then, according to the Figure 1, a set of shallow models including decision tree, random forest, linear regression, Naive Bayes and deep learning models including GRU, BLSTM and the combination of BLSTM-GRU models are used to predict the unknown network flow. In the next step, the model with the best performance is selected so that the coefficient of weights related to each feature is determined and the features with higher weight are found as the features that contributed the most to the performance of the model. In the Explanation step, an explanation is given about why these features caused the model to achieve high accuracy.

B. DATASET

The CICAndMal2017 dataset is one of the most complete network traffic datasets in the field of malware

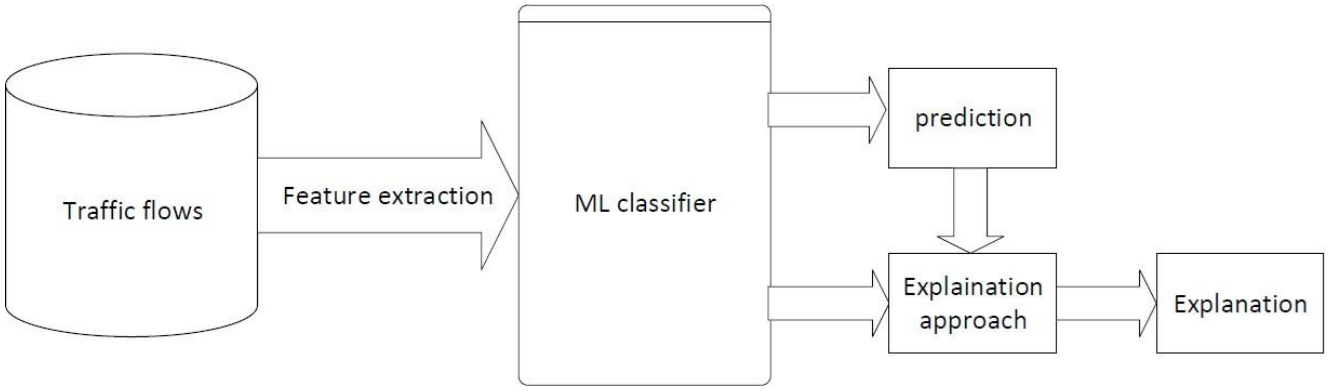

FIGURE 1: Overview of the proposed model.

detection. This data set includes four malware families, Adware, Ransomware, Scareware and SMS, each of which contains ten families, with a total of 40 families of different types of malware available, which are described in Table 1. In order to collect the collection of benign programs, applications that were produced in 2015, 2016 and 2017 and were very popular among users were used. According to the description of the dataset [25], The number of applications includes 1500 benign applications and 400 malware applications that were executed on a NEXUS 5 phone so that the phone was restarted after each run. The traffic flows were collected in a pcap format file. Each flow contains traffic packets that can be labeled as benign flow or malicious flow. The dataset consists of a large number of TCP flows, which contains valuable information from the sequences of traffic packets in each flow. Each record of the sequence represents a network flow and contains a vector of all the statistical features extracted from the network flow. This vector has 75 items as shown in Table 2. These features are based on four characteristics: byte-based, packet-based, time-based, and flow -based. Some features can describe specific (unknown) attacks and some features can be used to describe other common attacks. For example, most threats involve a wide range of remote system intrusions, usually detected by the irregular sent / received of packets on the network. To detect a specific attack, a deep learning model must be designed that discovers the relationships between traffic data in different representations.

C. EXPLAINABLE MODELS

Machine learning is a data analysis method that automatically performs analytical modeling. In this article, with the aim of identifying Android malware, we intend to investigate the impact of machine learning methods in more accurately identifying the presence of malware in smartphones. These models can provide explanations of prediction results and information in a way that is understandable for humans. The Defense Advanced Research Projects Agency (DARPA) coined the term "explainable artificial intelligence" (XAI). It is called "white box" because it explains the process of the model [26]. XAI approaches make it possible for users to be confident about the output results and based on the knowledge inferred from the descriptions, the accuracy of the results can be improved. Explainable artificial intelligence methods are classified into two categories: local explanation and global explanation methods.

- Local: local models provide specific, instance-level explanations for individual predictions. These models describe the capacity of a system to show the reason for a particular choice or decision to the user and is emphasized as the first important component of model transparency [22], [27] . Some local explainability methods such as LIME [28], SHAP [29] and counterfactual explanations [30] are common methods in this category.
- Global: models provide a high-level understanding of how your AI model is making predictions. global explainability refers to the explanation of the learning algorithm as a whole, taking into account the training data used, appropriate applications of the algorithms, and any caveats about shortcomings and incorrect applications of the algorithm.

There are several basic XAI techniques for producing explanations that are both accurate and understandable. One of the techniques used is the feature engineering technique. In this method, the most important features of the input that played the most important role in deciding the learning model are determined. In this article, a set of deep and shallow learning models, including decision tree, random forest, linear regression, Naive Bayes and GRU, LSTM, BLSTM and the combination of BLSTM-GRU neural networks are compared and the model with higher accuracy is selected. Then the features that were very influential in the decision making process of the model are selected.

TABLE 1: Overview of malware Statistics

	TYPE	NO of samples
ADWARE	Dowgin	10
	Evind	10
	Feiwo	15
	Gooligan	11
	Kemoge	10
	Koodous	10
	Mobidash	10
	Selfmit	4
	Youmi	10
	Shuanet	10
RANSOMWARE	Charger	10
	Jisut	10
	Koler	10
	Lockerpin	10
	Pletor	10
	Porndroid	10
	Ransombo	10
	Simplocker	10
	SVpeng	10
	Wannalocker	10
SCAREWARE	AndroidDefender	14
	AndroidSpy	6
	AVforAndroid	10
	AVpass	10
	FakeApp	10
	FakeAppal	11
	FakeAV	10
	Penetho	10
	FakeJobOfer	9
	VirusShield	10
SMSMALWARE	Beanbot	9
	Biige	11
	Fakeinst	10
	FakeMart	10
	FakeNotify	10
	Jifake	10
	Mazarbot	9
	Nandrobox	11
	Plankton	10
	Zsone	10

TABLE 2: Lists of network features

Byte-based features
Average number of bytes sent (received)
The total number of bytes used for headers sent (received)
Ratio of number of incoming bytes to number of outgoing bytes
Average number of bytes per second
Packet-based
Total number of packets sent (received)
Total length of packets sent (received)
Average number of packets per second
Average number of packets sent (received) per second
Min, Mean, Max, and standard deviation of the size of packet
Min, Mean, Max, and standard deviation of the size of packet sent (received)
Average number of packets sent (received)/Bulk
Sub-flow packets sent (received)
Ratio number of Incoming packets to number of Outgoing packets
Time-based features
Min, Mean, Max, and standard deviation time between two packets sent in the forward (backward) direction
Min, Mean, Max, and standard deviation time a flow was idle before becoming active (idle)
Flow-based features
The duration of the flow
Min, Mean, Max, and standard deviation of the length of a flow
Average number of packets per flow
Average number of packets sent (received) per flow
Average number of bytes sent (received) per flow
The average number of bytes in a sub flow in the forward (backward) direction
Variance of total number of bytes used in the forward (backward) direction
Number of packets with FIN, SYN, RST, PSH, ACK, URG, CWE
The total number of bytes sent in initial window in the forward (backward) direction
Ave, Max/Min segment size observed in the forward (backward) direction

IV. EXPERIMENTS AND DISCUSSION

In this section, we examine and compare each of the learning models in order to identify unknown malware. For this purpose, we use several criteria to measure the performance of the model. These criteria include accuracy, precision, recall and F1-Score.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - \text{Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

In the formulas above, TP is the number of positive samples that are correctly classified. FP is the number of negative samples that are misclassified. FN is the number of falsely classified positive samples. and TN is the number of negative samples that are correctly classified. precision, the ratio of the number of true malicious samples that are correctly classified to the total number of all samples that are classified as malicious. Recall is the ratio of malicious samples that are correctly classified to the total number of malicious samples that are correctly classified as malicious or falsely classified as malignant. F1-Score is the balanced average between recall and precision. In this section, in order to examine the performance of learning models in detecting unknown malware, we need to train the learning models with samples that have been trained to date and evaluate how the model performs with samples that have not been seen so far. To do this, we consider one year as the dividing point between the unknown data and the trained data. We use samples from the same year and previous years to train the model, and

we use samples after the separation point as test data that have not been seen before. Therefore, the model is tested with a new unknown dataset. To implement deep learning models, libraries based on deep learning tools, including the Keras library and the Tensorflow tool, were used in the Python environment. In order to improve the performance of deep learning models in detecting unknown malware, GRU, LSTM and BLSTM methods and the combination of BLSTM-GRU models were used. In this research, in order to eliminate the adverse effects caused by heterogeneous data, it is necessary to standardize the data in such a way that they fall within the range of [0, 1]. So first we normalized the data. For the GRU model, we applied the designed model on the data set. The structure of the GRU neural network in this research is such that the data records are read one by one and all the features in each record are assigned to a memory block. The model has 50 connected memory blocks, each block consists of two memory cells. The values of each record read as the first input are entered into the cells within that block, and since the cells are connected, the output of each cell is entered into the next cell in the same block at each time step. All steps continue until the last block and the output of the last block is considered as the final output of the model. The connection between each memory block for data that is sequentially related to each other causes the next records to use the state of the previous records in determining the output of each block, which is effective in predicting the model more accurately. The parameters in this model were set to have the least error. Therefore, to implement the GRU network, the keras library with two layers of GRU and Adam was used as an optimizer. The experiment was analyzed for 10 epochs and Batch size of 50 and in 50 time steps. In the LSTM model, all features are entered into the memory block in each record. There are 50 memory blocks in each layer and each block contains two cells. Each record is entered into the cells of that block as the first input, and the output of the last block is calculated as the output of the layer. To implement the BLSTM model, two layers are used, the upper layer calculates the output of the forward hidden layer from time step 1 to time step 50 and obtains and stores the output at any time. The lower layer is calculated inversely from time step 50 to time step 1, and the output of the backward hidden layer is obtained and stored each time. Then the values of two layers are averaged. The combined BLSTM-GRU method also includes two layers of BLSTM and two layers of GRU. In the third and fourth layer, the output sequence of the previous layers is entered as input in 50 memory blocks in series, and the output of the last block in the last layer is considered as the final output. To implement the combined method, Adam is used as an optimizer and sigmoid function is used as an activity function. The optimal number of parameters

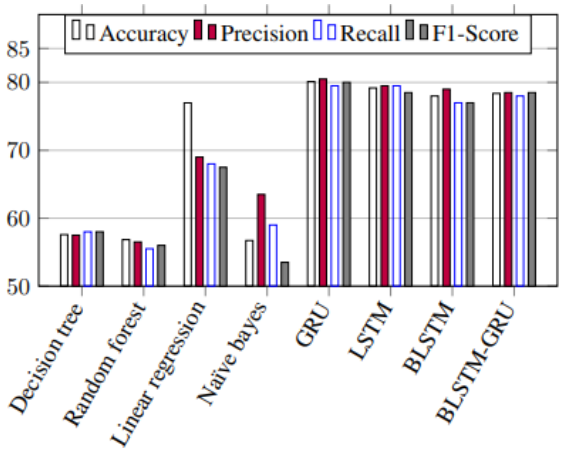

FIGURE 2: Results for detection of unknown malware from 2015.

is determined by experiment. Therefore, the value of 50 epochs and the value of 50 Batch size are determined for the implementation of the model.

A. EVALUATION RESULTS

The evaluations for the years 2015 to 2017 are shown in Figure 2 to Figure 4. In these evaluations, the detection of the type of malware family is omitted, and the classes include malware and benign, and all malware samples from all families in Malware classes are classified. According to the obtained results, BLSTM and BLSTM-GRU models performed better than other methods in detecting unknown traffic for 2016 and 2017, and GRU and LSTM models performed better in 2015. The highest accuracy with a high value of 89% is related to the BLSTM and BLSTM-GRU models for the data of 2017, which is a significant value. Also, in 2017, all models have a higher accuracy in detecting unknown malware compared to the previous two years, which is one of the reasons for training the model with a higher volume of malware traffic, which causes the model to be trained with more diverse patterns of malware, and as a result can predict better. Considering that BLSTM and BLSTM-GRU models were able to predict unknown traffic with an accuracy of about 90%, therefore these two models are used to find robust and effective features.

B. EVALUATION OF EXPLAINABLE ARTIFICIAL INTELLIGENCE

In the next step of the evaluation, it is discussed how deep learning models make decisions that make the model perform well and according to the results of the explanation, upgrade the model and improve the performance of the model. In order to understand the model's decision-making, explanations are necessary to determine which part of the network's features had the greatest impact in predicting unknown attacks. There-

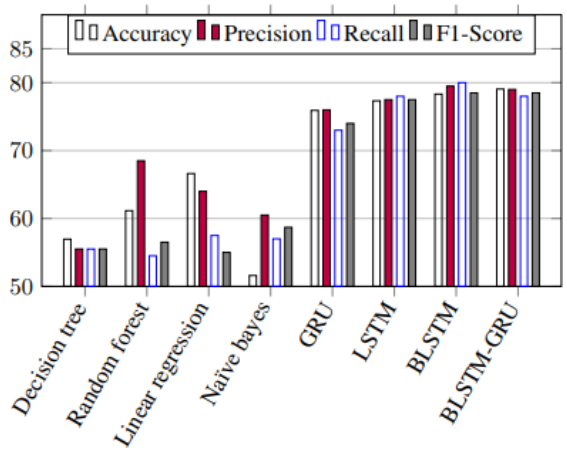

FIGURE 3: Results for detection of unknown malware from 2016.

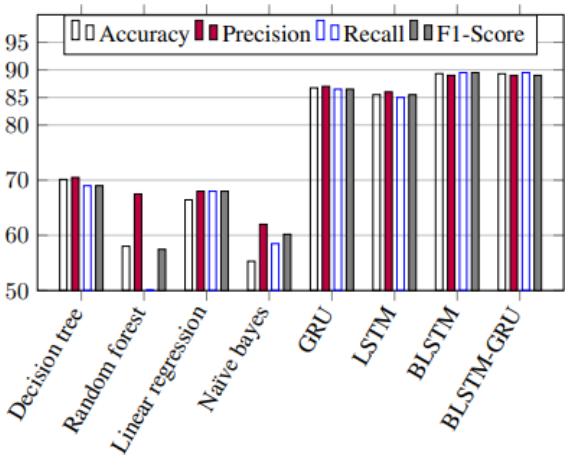

FIGURE 4: Results for detection of unknown malware from 2017.

fore, the SHAP method is used as one of the explanatory methods to find important features. Figure 5 shows the importance of all 77 features in brief using the SHAP method for the BLSTM-GRU hybrid model. The high values of each bar chart represent a greater contribution to the output of the model and indicate the high importance of that feature in the output of the model. Lower values also contributed less to model prediction. According to the figure, feature "Fwd IAT Total" had the greatest impact in predicting the BLSTM-GRU model and then the features of “Fwd Packet Length Max”, “Max Packet Length”, “Fwd IAT Std” and “Flow IAT Max” respectively had a high impact factor in the output of the model. Figure 6 also shows SHAP values for BLSTM algorithm. In this model, feature " Fwd Packet Length Max " had the greatest impact in predicting the model. Features “Fwd IAT Total”, “Max Packet Length”, “Flow IAT Max”, “Average Packet Size”, were also robust features for BLSTM algorithm respectively. Some features such as "Fwd IAT Total", “Max Packet Length”, “Fwd Packet Length Max” and “Flow IAT Max” in both BLSTM and BLSTM-GRU methods had a great role in predicting the learning model. Therefore, it can be concluded that these four features were very robust.

C. RESULTS AND DISCUSSIONS ABOUT EFFECTIVE FEATURES

In the previous subsection, the effective features in the detection of new malware were examined and a number of robust features were identified. The features of "Fwd IAT Total", "Max Packet Length", "Fwd Packet Length Max" and "Flow IAT Max" were robust features with high impact factor in both BLSTM and BLSTM-GRU methods. This sub-section discusses the use of robust features in differentiating between malware and benign traffic, as well as whether the impact of these features will be reduced by changing the behavior pattern of malware.

- fwd IATtotal:In this feature, IAT means (Inter Arrival Time) refers to the time interval between two consecutive messages or data packets. This feature indicates the total time interval between inputs (messages or data packets) of a system or network. In other words, this feature indicates the sum of the delay times between consecutive inputs. A malware may manipulate and falsify information about delay times between inputs in order to use it to send malicious or fake data. These types of changes may interfere with the detection of malware traffic.
- Max Packet Length:This feature refers to the length of the most data packets sent in a certain period of time. This feature can be useful for malware detection because malware may send data packets of very large or very small lengths that differ from normal network traffic patterns. On the other hand, malware may use this feature to hide their activities or prevent them from being detected. For example, a malware may send data packets of very small length to avoid detection by security systems. This method is known as a type of evasion technique because the data packets with smaller length may be lost among the network traffic and simply not detected by malware detection systems. Using this method, malware tries to disrupt the detection of specific patterns of network traffic and challenge security systems.
- Fwd Packet Length Max :This feature shows the largest length of data packets sent to the destination. In other words, this feature shows the maximum length of data packets that have been sent forward from a device. The main difference between this feature and the Max Packet Length

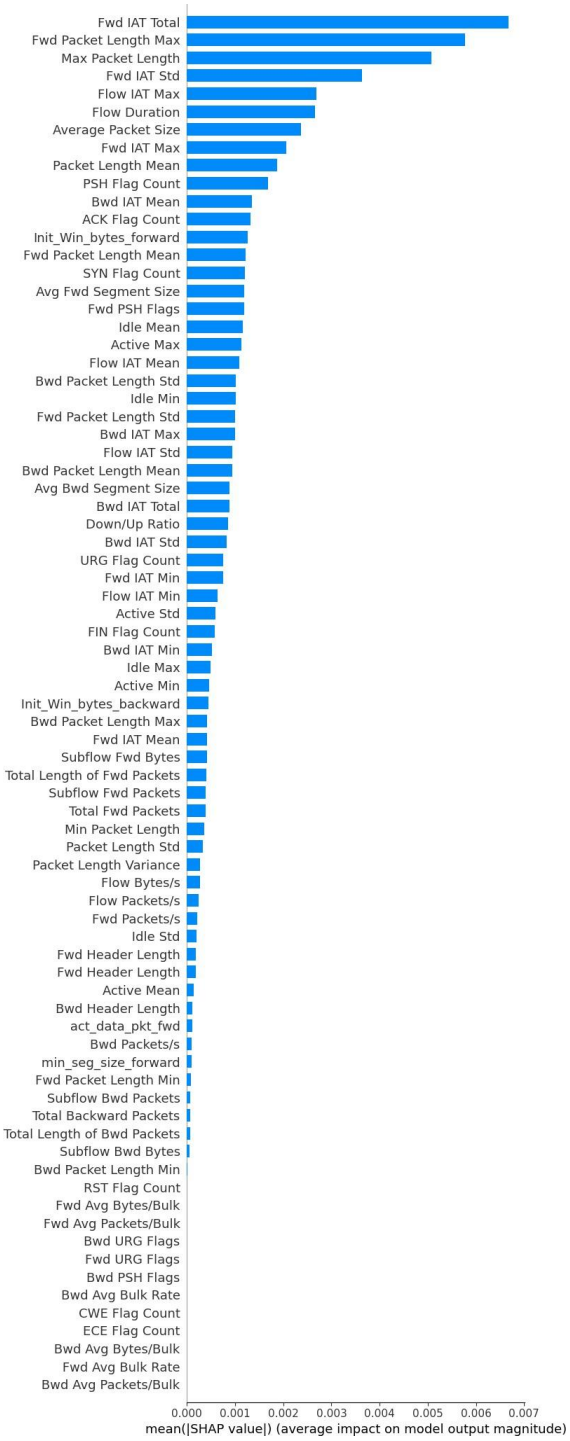

FIGURE 5: SHAP values for BLSTM-GRU model .

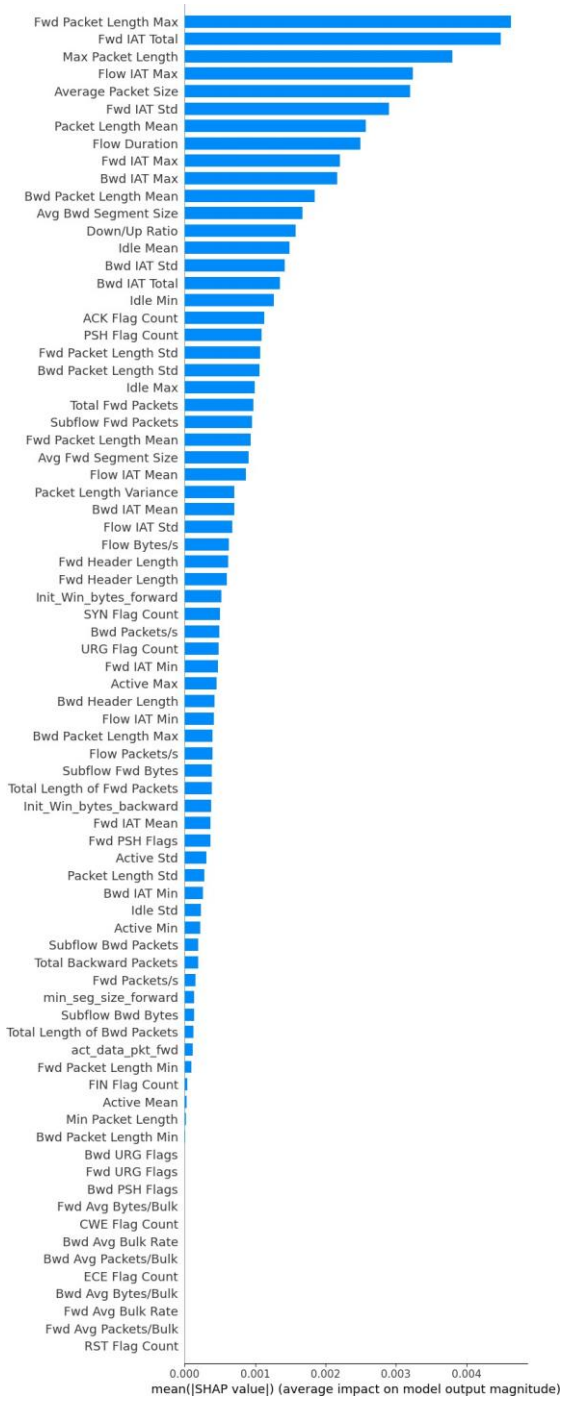

FIGURE 6: SHAP values for BLSTM model .

feature is that the Fwd Packet Length Max feature shows the maximum length of data packets from a system forward, while Max Packet Length shows the maximum length of data packets of the entire network traffic. In this feature, like the Fwd Packet Length Max feature, malware may try to avoid detection by malware detection systems by sending shorter data packets or variable length data packets.

- Flow IAT Max :This feature shows information about the maximum delay time between two consecutive data packets sent in a flow. The longest delay time between two consecutive data packets may indicate unusual patterns in data transmission by malware or malicious programs. In other words, malware may try to use time delays between data transmissions in order to secretly increase their activity. Malware may change the maximum delay time information between two consecutive data packets to avoid detecting unusual patterns in network traffic.
- fwd IAT std : This feature means the standard deviation of the time between two consecutive forward packets. This feature shows the variance or dispersion of time between two consecutive packets sent from a device forward in the network. In other words, this feature shows how much the time between two consecutive packets may differ. Malware may try to change the time between sending packets or add fake packets to change the standard deviation of the time between packets. In other words, malware may try to avoid detecting and preventing their activities by security systems in the network by changing the pattern of sending packets.
- Average packet size : This feature means the average size of data packets sent from one device to another in the network. This feature shows the average amount of data that is in each sent and received packet. In other words, this feature shows how much data each packet carries on average. Malware may change the average packet size by adding fake data to the packets or changing the size of the packets.
- flow duration: This feature means the duration of a data flow between two devices or nodes in the network. In fact, this feature indicates the time that a data stream is continuously transmitted in the network from the start time to the end time. This feature can be useful for analyzing network usage patterns, detecting attacks and intrusions, or improving network performance. Malware may try to change the duration of the data flow in order to hide the pattern of its activities or avoid detection and prevention by security systems in the network.
- packet length mean: This feature means the average length of data packets that are transmitted between two devices or nodes in the network. In other words, this feature indicates the average length of data packets that are transmitted in a data flow between two points in the network. A malware may try to change the length of data packets in order to hide its activity pattern.
- fwd IAT max: This feature indicates the maximum time that passes between two packets sent consecutively (without interruption) from a device or node forward in the network. Malware may try to change the time between the arrival of two consecutively sent packets to hide its activity pattern.

By analyzing and examining the effective features of network traffic and other related features , it can be concluded that no specific features of network traffic can be definitely not manipulated by malware. Malware attacks can affect the values of network traffic features and their nature. Attacks through different methods can change and distort the characteristics of network traffic, which are mentioned in a few cases:

- Changing the traffic pattern: Malware usually have specific patterns in sending and receiving data. For example, they may send data packets at special and unusual time intervals that differ from the normal behavior of normal traffic.
- Hiding their activities: Malware may try to hide their activities and use changes in network traffic features to avoid detection and interception by network security systems.
- Destruction and disruption of information: By changing and distorting the features of network traffic, malware may carry out infiltration and espionage attacks on the network and steal sensitive information or control devices.
- Intrusion and espionage attacks: Malwares may perform network intrusion and espionage attacks by changing and distorting network traffic features and steal sensitive information or control devices.
- Denial of Service (DOS) attacks: By increasing or decreasing the volume of traffic, malware may carry out denial of service attacks on the network and cause communication interruptions and server malfunctions.

However, malware detection systems can identify specific patterns of malware traffic and detect and block malware more effectively. In addition, the use of Deep Packet Inspection methods and artificial intelligence algorithms to detect unusual patterns and intrusion attacks can also be effective.

V. CONCLUSIONS

In this article, traffic data was used to predict unknown malware and data processing was fed to learning models as a sequence of traffic flows. In addition, in this article, a new approach to detect unknown malware based on explainable artificial intelligence was proposed. This study

applied several learning algorithms including decision tree, random forest, linear regression, Naive Bayes and deep learning models including GRU, LSTM, BLSTM and the combination of BLSTM-GRU models to the data and the performance of the models in detecting unknown malware with were compared to each other. According to the evaluation criteria, BLSTM algorithms and combined BLSTM-GRU algorithm performed better than others. This paper also used XAI and tried to explain artificial intelligence through SHAP to prove the validity of the method. According to the explanations obtained, it is possible to obtain information about what the learning models have learned from the features and whether the tested deep models have reliable reasons for their decisions or follow simple or irrelevant patterns. In the future work, we plan to check whether the performance of the model can be increased by adding more complex features or changing some of the feature parameters, according to the explanations generated from XAI.

REFERENCES

[1] A. Abusitta, M. Q. Li, and B. C. Fung, "Malware classification and composition analysis: A survey of recent developments," Journal of Information Security and Applications, vol. 59, p. 102828, 2021.

[2] S. Morgan, "Cybercrime to cost the world 8 trillion annually in 2023," Available online: https://cybersecurityventures.com, 2022.

[3] A. Demontis, M. Melis, B. Biggio, D. Maiorca, D. Arp,
K. Rieck, I. Corona, G. Giacinto, and F. Roli, "Yes, machine learning can be more secure! a case study on android malware detection," IEEE transactions on dependable and secure computing, vol. 16, no. 4, pp. 711–724, 2017.

[4] J. Yan, Y. Qi, and Q. Rao, "Lstm-based hierarchical denoising network for android malware detection," Security and Com- munication Networks, vol. 2018, pp. 1–18, 2018.

[5] F. Charmet, H. C. Tanuwidjaja, S. Ayoubi, P.-F. Gimenez,
Y. Han, H. Jmila, G. Blanc, T. Takahashi, and Z. Zhang, "Ex- plainable artificial intelligence for cybersecurity: a literature survey," Annals of Telecommunications, vol. 77, no. 11-12,
pp. 789–812, 2022.

[6] S. Fallah and A. J. Bidgoly, "Android malware detection using network traffic based on sequential deep learning models," Software: Practice and Experience, vol. 52, no. 9, pp. 1987– 2004, 2022.

[7] ——, "Benchmarking machine learning algorithms for android malware detection," Jordanian Journal of Computers and Information Technology, vol. 5, no. 3, 2019.

[8] J.-Y. Kim and S.-B. Cho, "Obfuscated malware detection using deep generative model based on global/local features," Computers & Security, vol. 112, p. 102501, 2022.

[9] X. Xing, X. Jin, H. Elahi, H. Jiang, and G. Wang, "A malware detection approach using autoencoder in deep learning," IEEE Access, vol. 10, pp. 25 696–25 706, 2022.

[10] C. Li, Q. Lv, N. Li, Y. Wang, D. Sun, and Y. Qiao, "A novel deep framework for dynamic malware detection based on api sequence intrinsic features," Computers & Security, vol. 116,
p. 102686, 2022.

[11] M. Asam, S. H. Khan, A. Akbar, S. Bibi, T. Jamal, A. Khan,
U. Ghafoor, and M. R. Bhutta, "Iot malware detection architecture using a novel channel boosted and squeezed cnn," Scientific Reports, vol. 12, no. 1, p. 15498, 2022.

[12] S. R. T. Mat, M. F. Ab Razak, M. N. M. Kahar, J. M. Arif,
and A. Firdaus, "A bayesian probability model for android malware detection," ICT Express, vol. 8, no. 3, pp. 424–431, 2022.

[13] R. Chaganti, V. Ravi, and T. D. Pham, "Deep learning based cross architecture internet of things malware detection and classification," Computers & Security, vol. 120, p. 102779, 2022.

[14] K. Shaukat, S. Luo, and V. Varadharajan, "A novel deep learning-based approach for malware detection," Engineering Applications of Artificial Intelligence, vol. 122, p. 106030, 2023.

[15] O. J. Falana, A. S. Sodiya, S. A. Onashoga, and B. S. Badmus, "Mal-detect: An intelligent visualization approach for malware detection," Journal of King Saud University- Computer and Information Sciences, vol. 34, no. 5, pp. 1968– 1983, 2022.

[16] B. Urooj, M. A. Shah, C. Maple, M. K. Abbasi, and S. Ri- asat, "Malware detection: a framework for reverse engineered android applications through machine learning algorithms," IEEE Access, vol. 10, pp. 89 031–89 050, 2022.

[17] J. H. Jimenez and K. Goseva-Popstojanova, "Malware detection using power consumption and network traffic data," in 2019 2nd International Conference on Data Intelligence and Security (ICDIS). IEEE, 2019, pp. 53–59.

[18] Y. Wu, J. Shi, P. Wang, D. Zeng, and C. Sun, "Deepcatra: Learning flow-and graph-based behaviours for android malware detection," IET Information Security, vol. 17, no. 1, pp. 118–130, 2023.

[19] S. Hariharan, A. Velicheti, A. Anagha, C. Thomas, and
N. Balakrishnan, "Explainable artificial intelligence in cyber- security: A brief review," in 2021 4th International Confer- ence on Security and Privacy (ISEA-ISAP). IEEE, 2021, pp. 1–12.

[20] N. Capuano, G. Fenza, V. Loia, and C. Stanzione, "Explain- able artificial intelligence in cybersecurity: A survey," IEEE Access, vol. 10, pp. 93 575–93 600, 2022.

[21] F. Ullah, A. Alsirhani, M. M. Alshahrani, A. Alomari,
H. Naeem, and S. A. Shah, "Explainable malware detection system using transformers-based transfer learning and multi-model visual representation," Sensors, vol. 22, no. 18, p. 6766, 2022.

[22] Z. Zhang, H. Al Hamadi, E. Damiani, C. Y. Yeun, and
F. Taher, "Explainable artificial intelligence applications in cyber security: State-of-the-art in research," IEEE Access, 2022.

[23] Y. Liu, C. Tantithamthavorn, L. Li, and Y. Liu, "Explainable ai for android malware detection: Towards understanding why the models perform so well?" in 2022 IEEE 33rd In- ternational Symposium on Software Reliability Engineering (ISSRE). IEEE, 2022, pp. 169–180.

[24] S. Gulmez, A. G. Kakisim, and I. Sogukpinar, "Xran: Explainable deep learning-based ransomware detection using dynamic analysis," Computers & Security, vol. 139, p. 103703, 2024.

[25] A. H. Lashkari, A. F. A. Kadir, L. Taheri, and A. A. Ghor- bani, “Toward developing a systematic approach to generate benchmark android malware datasets and classification,” in 2018 International Carnahan conference on security technol- ogy (ICCST). IEEE, 2018, pp. 1–7.

[26] A. Saranya and R. Subhashini, “A systematic review of explainable artificial intelligence models and applications: Recent developments and future trends,” Decision analytics journal, p. 100230, 2023.

[27] V. Arya, R. K. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, A. Mojsilović et al., “One explanation does not fit all:

[30] S. Wachter, B. Mittelstadt, and C. Russell, “Counterfactual explanations without opening the black box: Automated

A toolkit and taxonomy of ai explainability techniques,” arXiv preprint arXiv:1909.03012, 2019.

[28] M. T. Ribeiro, S. Singh, and C. Guestrin, “" why should i trust you?" explaining the predictions of any classifier,” in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.

[29] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” Advances in neural information processing systems, vol. 30, 2017.

decisions and the gdpr,” Harv. JL & Tech., vol. 31, p. 841, 2017.

HUSSEIN ALI GHADHBAN SALMAN received his B.Sc. degree in Computer Science from Iraq University of Basrah in 2007, M.Sc. in Information System from India Osmania University in 2015 and Ph.D in Computer Engineering from Iran University of Qom, Currently. His researches deals with the field of computer science, information systems and information technology.

AMIR JALALY BIDGOLY received his M.Sc. degree in Software Engineering from Iran University of Science and Technology (IUST) in 2009, and Ph.D in Software Engineering from University of Isfahan (Isfahan, Iran) in 2015. He is currently an assistant professor with the Department of Computer Engineering at University of Qom. His research interests include computer security and machine learning (deep learning).

SOMAYYEH FALLAH received his M.Sc. degree in Information Technology from University of Qom (Qom, Iran). Her current research interests are research with deep learning methods. She has done several researches in different fields such as malware detection, prediction of stock, etc. with deep learning methods.