Improving Automated Labeling for ATT&CK Tactics in Malware Threat Reports

EVA DOMSCHOT and RAMYAA RAMYAA, New Mexico Institute of Mining and Technology, USA
MICHAEL R. SMITH, Sandia National Laboratories, USA

Once novel malware is detected, threat reports are written by security companies that discover it. The reports often vary in the terminology describing the behavior of the malware making comparisons of reports of the same malware from different companies difficult. To aid in the automated discovery of novel malware, it was recently proposed that novel malware could be detected by identifying behaviors. This assumes that a core set of behaviors are present in most, if not all, malware variants. However, there is a lack of malware datasets that are labeled with behaviors. Motivated by a need to label malware with a common set of behaviors, this work examines automating the process of labeling malware with behaviors identified in malware threat reports despite the variability of terminology. To do so, we examine several techniques from the natural language processing (NLP) domain. We find that most state-of-the-art word embedding NLP methods require large amounts of data and are trained on generic corpora of text data—missing the nuances related to information security. To address this, we use simple feature selection techniques. We find that simple feature selection techniques generally outperform word embedding methods and achieve an increase of 6% in the $F_{.5}$-score over prior work when used to predict MITRE ATT&CK tactics in threat reports. Our work indicates that feature selection, which has commonly been overlooked by sophisticated methods in NLP tasks, is beneficial for information security related tasks, where more sophisticated NLP methodologies are not able to pick out relevant information security terms.

CCS Concepts: • **Computing methodologies** → **Feature selection**; *Supervised learning by classification*; • **Security and privacy** → Malware and its mitigation;

Additional Key Words and Phrases: Machine learning, feature selection, malware detection, natural language processing, cybersecurity, MITRE ATT&CK

ACM Reference format:
Eva Domschot, Ramyaa Ramyaa, and Michael R. Smith. 2024. Improving Automated Labeling for ATT&CK Tactics in Malware Threat Reports. *Digit. Threat. Res. Pract.* 5, 1, Article 2 (March 2024), 16 pages.
https://doi.org/10.1145/3594553

Authors' addresses: E. Domschot and R. Ramyaa, New Mexico Institute of Mining and Technology, Socorro, NM; emails: eva.domschot@student.nmt.edu, ramyaa@cs.nmt.edu; M. R. Smith, Sandia National Laboratories Albuquerque, NM; email: msmith4@sandia.gov.

2576-5337/2024/03-ART2 $15.00
https://doi.org/10.1145/3594553

1 INTRODUCTION

Malware continues to represent a significant threat to information security, causing trillions of dollars of damage every year. It is estimated that 2.9 million dollars is lost every minute from the global economy due to cyber-attacks [26]. A principal problem is that traditional ways of defining malware are very limited in scope. Until recently, antivirus companies like Comodo, Kaspersky, Kingsoft, and Symantec have provided software to detect malware that relies on signature-based methods [36]. These signatures are short sequences of bytes that are unique to each malware executable, which produces brittle malware detectors with little capability of classifying new malware files. However, by the time they identify the new signature as malicious, malware generally has a large infection rate and often has already morphed into a new variant (polymorphic malware) [2]. The average length of time it takes to identify the new malware samples with traditional detection techniques is around 54 days, and it is estimated around 15% of malware is still undetected after 180 days [36].

To fix the above problem, a concerted effort by the research community has resorted to using **machine learning (ML)** methods to detect malware. These methods follow a process of feature extraction of the malware executable, followed by classification or clustering of the malware samples [36]. Ye et al. provide a survey of ML approaches and show that many ML methods achieve classification rates well above 90% for identifying malware samples. However, while the ML methods do quite well on samples of malware they have already learned, these models are often easily evaded in practice [32]. Further complicating the issue of detecting malware with ML is adversarial learning. Adversarial learning allows attackers to perturb malware samples enough so that when they are inputted into machine learning models, the models are fooled into labeling them as benign [12]. Many of these adversarial samples are examples of polymorphic malware, and it is not well understood why these models are so easily misled [12].

Recent work has began to identify new ways in which zero-day malware can be identified without relying on signatures and instead working to identify malicious behaviors [32]. Here behaviors refer to actions taken by the malware such as **dynamic link library (DLL)** injection or process hollowing. Additionally, while behavioral analysis in information security typically refers to dynamic analysis, here we mean the labels that a malware detector predicts and can be used with inputs from static or dynamic analyses. This work focuses on labeling malware samples with behavioral labels through the use of malware threat reports following Smith et al. [32]. Malware threat reports are reports written by malware analysts, giving in-depth descriptions of how different kinds of malware behave. Unfortunately, there is often inconsistency between different authors and organizations in how these reports are written and terms that are used, making the process of labeling malware from them a manual, laborious task. Describing the actions taken by malware would significantly reduce analyst time in triaging an attack. Thus, labeling malware samples with behaviors from threat reports would provide additional methods to detect malware.

In 2020, Legoy et al. used a dataset of malware threat reports that had been labeled with MITRE ATT&CK tactics and techniques [23]. As this is a multi-label dataset, each label is given an individual classifier. To summarize the results, they reported both the micro and macro averages of the precision, recall, and $F_{.5}$-score, averaging over the results for every label. The $F_{.5}$-score represents the weighted average between the recall and the precision, with the precision being weighted as twice as important as the recall. The $F_{.5}$-score was reported instead of the F1-score so that more importance is given to assigning labels correctly (precision), even though some labels may not be assigned (recall) [23]. The macro average is a standard average, where every label is treated equally [14]. In contrast, the micro average weights each sample equally, meaning it is able to capture any imbalance of the distribution between labels. A macro average may overestimate or underestimate the average if some of the labels have many more positive samples than other labels [14]. However, because the micro average does not weight every label equally, if one label has many samples and those samples are predicted correctly, then the classifier will be seen as doing well even if the classifier has poor prediction accuracy for a label with fewer samples. The micro average is best used if it is not important how well the classifier is doing on predictions for each label. For this reason, we have chosen to only use the macro average, because we want to weight each tactic

Table 1. Tactics Breakdown

Tactics	Number of Reports
Credential Access	306
Execution	488
Impact	75
Persistence	590
Privilege Escalation	403
Lateral Movement	343
Defense Evasion	768
Exfiltration	123
Discovery	356
Collection	236
Command and Control	406
Initial Access	202

label equally, so that we know whether the classifier is performing well on all of the labels rather than potentially only performing well on labels that have more samples. This differs from other ML problems not dealing with multi-label or multi-class datasets. In summary, the work done by Legoy et al. illustrates the challenges with using **natural language programming (NLP)** techniques that are not specific to the information security domain, their best model achieving a macro average $F_{.5}$-score of 27.52% for techniques and 59.47% for tactics.

The MITRE ATT&CK framework is a knowledge base of adversary tactics and techniques, taken from observations of attackers and malware [23]. In recent years it has gained prominence as a way of creating a standardized language around discussing cyber threats. The MITRE ATT&CK framework can be divided into three different different domains, namely Enterprise, Mobile, and ICS. Enterprise describes behaviors on standard IT systems, such as Linux or Windows; Mobile describes behaviors on mobile devices, using operating systems such as Android; and ICS describes behaviors carried out against industrial control system networks. The Enterprise domain, on which this work focuses, can be divided into tactics and techniques. Tactics represent the goal of the attacker or malware sample, and a list of the 12 MITRE ATT&CK tactics is given in Table 1. Every tactic has multiple techniques, which are ways in which the tactic can be achieved. In this work, the behaviors align with the techniques. Despite the creation of the MITRE ATT&CK framework, its adoption has yet to be universal.

The work in this article focuses on automating the process of labeling malware threat reports follwing the MITRE ATT&CK framework and improving the results of Legoy et al. This work is an important step toward labeling malware datasets with information that can be extracted from malware threat reports. Our contributions to this task include an examination of word embedding techniques that have commonly been used for NLP tasks, as well as feature selection methods that are not commonly used as a part of the pipeline of NLP related tasks. We demonstrate that current word embedding methods that are not domain specific are insufficient for information security tasks. In response, we suggest the usage of basic feature selection approaches, which show improved performance in the labeling of MITRE ATT&CK tactics in threat reports. This work indicates that, although feature selection is not used as commonly for NLP tasks, it may be useful for information security tasks, where more sophisticated methods are not able to pick out the relevant domain-specific terms. We improve over the work done by Legoy et al., who used non-domain-specific NLP methods. Our work shows an $F_{.5}$-score of 65% for the prediction of tactics when using feature selection techniques to create a feature set of the most relevant information security terms for each label, compared to 59% from Legoy et al. All of the code used for this project can be found at our Git repository.[1]

[1] https://gitlab.com/malgen/behavior_labels.

2 RELATED WORK

In 2017, Lim et al. [24], created an annotated set of malware threat reports by hand, using the **Malware Attribute Enumeration and Characterization (MAEC)** vocabulary, a system defining malware behaviors, malware capabilities, malware families, malware instances, and malware collections. In the end, they created a labeled dataset of 39 malware threat reports by mapping malware actions and behaviors to MAEC labels. They then used a **support vector machine (SVM)** and naïve Bayes to predict these MAEC attribute labels, achieving an F1-score of 40% for identifying malware capabilities, which were the most successfully predicted. Malware capabilities most closely align with MITRE ATT&CK tactics, with equivalent labels such as exfiltration and privilege escalation. Overall, this work illustrates the difficulty of only having a small labeled dataset of 39 documents that has many more labels than it does samples.

In 2020, Legoy et al. [23], attempted to automate the prediction of MITRE ATT&CK tactics and techniques on a dataset of malware threat reports. Their work differed from the work of Lim et al. who annotated malware threat reports by hand. Instead, they used MITRE ATT&CK tactics and techniques to create a labeled dataset. To do this, they used the references to threat reports that MITRE ATT&CK has for each tactic and technique. To classify these documents, Legoy et al. tested many different techniques including regularization techniques, which included a Ridge Regression Classifier; multiple tree based methods, such as Random Forest ensembles; and different kinds of boosting algorithms. They also explored a multi-label classification method, classifier chains, that makes the assumption that there are correlations between labels. Their best-performing method was a simple **Linear SVM (L. SVM)** classifier using **term frequency-inverse document frequency (TF-IDF)**, a **bag-of-words (BOW)** representation that ensures that terms that appear across the majority of documents have a lower weighted representation [18]. For the prediction of tactics, they obtained a macro average precision score of 60.26%, a macro average recall score of 58.5%, and an $F_{.5}$-score of 59.47%. Our work attempts to improve upon their work in two different ways. Their work did not explore the use of any deep learning methods combined with word embeddings to label the malware threat reports, which are state of the art for text classification tasks. Instead, their work mainly focused on testing classical machine learning methods, such as L. SVM, to make predictions. Our work explores the use of these deep learning methods combined with word embeddings on a domain-specific dataset. Second, we explored ways of improving upon their best-performing method (L. SVM) by introducing feature selection techniques to reduce the number of features, which exceeded the number of samples in the dataset, as discussed in Section 3. This was done to improve over-fitting and aid the classifier in choosing the relevant terms [40]. We also explored methods of handling imbalance in the dataset, which this dataset exhibited as discussed in Section 3.

In addition to linking malware threat reports to the MITRE ATT&CK framework, other work has been done to link **Common Vulnerabilities and Exposures (CVEs)** to the MITRE ATT&CK framework. The CVE list is maintained by MITRE corporation, who adds a new listing whenever a new security flaw is found in software or hardware [5]. These CVEs contain metadata that includes text descriptions of the vulnerability. In 2021, Ampel et al. used state-of-the-art transformer-based models to label CVEs with MITRE ATT&CK tactics. Their best results showed an F1-score of 76%, while classical machine learning methods did not achieve an F1-score higher than 48%. However, they noted that 91% of the data fell into four of the tactics and that some of the tactics such as Command and Control do not require vulnerabilities, meaning that the CVEs cannot be linked to them. For this reason, we have not chosen to utilize CVEs as part of our training set to label malware threat reports.

Additionally, another paper was published in 2021 by Kuppa et al., who sought to link CVEs to MITRE ATT&CK techniques. Kuppa et al. labeled CVEs by using joint embeddings with MLP. Using this method, they were only able to label 37 techniques indicating the difficulty in capturing all of the technique labels [22].

Another paper was published in 2022 by Andrew et al. [6] that sought to link the descriptions of Linux shell commands to MITRE ATT&CK tactics and techniques by using the cosine similarity between the shell command descriptions and the MITRE ATT&CK tactics and techniques descriptions, using both a TF-IDF representation and a Word2Vec representation. To do this, they labeled the shell commands with the top-n highest scoring

labels. Of note, Andrew et al. showed that TF-IDF performs better than Word2Vec on this task, highlighting the difficulty of using pre-trained word embeddings on a domain-specific task.

3 DATA

We examine feature selection and word embeddings for the labeling of behaviors defined by the MITRE ATT&CK framework to threat reports. The dataset comes from the rcATT repository, produced by Legoy et al.[2] The data contains 1,490 malware threat reports, labeled with MITRE ATT&CK tactics and techniques. The average length of the threat reports is 3,000 words, with some reports containing upwards of 60,000 words. Additionally, while the reports are labeled with the tactics and techniques that they contain, there are no metadata to show which sentences in a report are connected to which tactics and techniques, making it difficult to split the texts into chunks. This creates further challenges, since many NLP methods do not perform well on long text documents. While our end goal is to label the malware techniques in threat reports, this work focuses mainly on the ATT&CK tactics due to limited computing power. Given that there are 200 techniques, it would require feature selection methods to first be used over them, a task that could take several days to finish. However, this work is extendable to the techniques as well, given more compute power. It should also be stated that feature selection only needs to be run once so long as the feature sets are stored, making the issue of processing speed a one-time problem. The threat reports come from a variety of sources, which means they do not have any explicit structure or consistent terminology. The data are also multi-label with 12 different labels for the tactics. Furthermore, this dataset is also imbalanced, with some tactics having as few as 75 samples, as shown in Table 1. Both class imbalance and multi-label problems create more difficult problems for ML.

For the TF-IDF methods, the data were pre-processed by removing all punctuation, non-letters, and **Natural Language Toolkit (NLTK)** [25] English stop words from the text. Additionally, the text was lemmatized, using the NLTK lemmatizer. For word embedding methods, the text was tokenized and either padded or cut into sequences of 3,000 words, because the majority of texts have an average length of 3,000 words or less. While some important information is likely lost from the texts that are longer than this length, this method allows us to retain the important information for the majority of the documents, while not maxing out computing resources. As there is no way to differentiate where important information is in the text documents, we have chosen to use the first 3,000 words of each text document. Each document is written in a different manner, such that key information seems to be left out regardless of whether the beginning, middle, or end of the threat reports is cut out for longer documents.

4 FEATURE SELECTION

Feature selection has been applied to text categorization as a way of improving performance, as well as the accuracy of a classifier [40]. Text often contains many words that are distractors or do not add any information within a document, reducing the overall accuracy of a classifier. As a result, feature selection methods are used to find an effective set of words to use as features that convey the most amount of information and remove distractor words. The two feature selection methods we used for this research, **mutual information (MI)** and the **Chi-squared statistic (CHI)**, are discussed below. For a broader discussion of possible NLP feature selection, see, for example, the survey presented by Kumbhar and Mali [21]. We focus on MI and the CHI statistics but other feature selection methods could have also been examined. We chose these two methods because of the simplicity and wide usage as representative feature selection methods.

4.1 Mutual Information

MI (I) measures the mutual dependence between two random variables and indicates how much can be determined about one variable by knowing the state of the other random variable. MI can be defined as shown in

[2] https://github.com/vlegoy/rcATT.

Equation (1), where t represents the term and c represents the category [34],

$$I(t,c) = log_2 \frac{P(t,c)}{P(t)P(c)}. \tag{1}$$

MI compares the joint probability of the term (t) and the category (c) occurring together with the probabilities of them occurring independently. When t and c are dependent, $P(t,c)$ should have a much higher probability than $P(t)P(c)$, and $I(t,c)$ should be greater than zero [34]. When independent, $I(t,c)$ is equal to zero. MI has been shown to have a bias toward terms with a low frequency [35].

MI has been shown to aid in feature selection with NLP tasks. MI was combined with a Bayesian algorithm to classify the topics of news documents, showing much improved results [27]. In addition, MI has been used for information security tasks. In the work by Amiri et al., they used MI to reduce the feature set for intrusion detection on network information [4]. MI did well on this task where intrusions are anomalous behaviors, perhaps due to MI picking out rare features. This is similar to the problem being focused on in this research where many labels are only conveyed in small portions of the text.

4.2 Chi-square Statistic

The CHI metric measures the independence between two events using statistical testing. The equation for the metric can be seen in Equation (2), where N is the number of documents, t is the term, c_i represents the ith category, $\bar{t}$ represents the absence of the term, and $\bar{c_i}$ represents non-membership of the category [40],

$$\chi^2(t,c_i) = \frac{N[P(t,c_i)P(\bar{t},\bar{c_i}) - P(t,\bar{c_i})P(\bar{t},c_i)]^2}{P(t)P(\bar{t})P(c_i)(\bar{c_i})}. \tag{2}$$

CHI is a normalized metric, which can be compared across features of the same category [35]. However, it is known that this normalization does not work well for low-frequency terms in the feature set, unlike MI, which places emphasis on rare terms [35]. CHI has been proven to do very well for text classification tasks, often outperforming other feature selection metrics [30].

4.3 Feature Selection for Imbalanced Datasets

As discussed in Section 3, the datatset we are using is imbalanced for many of the labels. Feature selection has been shown to improve results on imbalanced text datasets [40]. Instead of balancing the training data, feature selection metrics are used to choose the most informative words out of the text, obtaining an optimal set of features. Zheng et al. [40] showed that using feature selection that only focuses on the presence of the positive class does not achieve great benchmarks against not using feature selection. However, taking the absence of the class into consideration does have a noticeable impact in terms of the result. In addition, they note that two-sided feature selection methods such as CHI are not as good at this task as they seem. Two-sided feature selection methods are those that select features for both the positive and negative classes. In contrast, one-sided feature selection methods only take the features for the positive class into account, meaning that the classifier only has features that represent the presence of a label. These methods tend to be biased toward terms correlated with the positive label, especially when the dataset is imbalanced, because the positive features occur less in the dataset and are usually scored more highly due to that fact. Instead, they found that one-sided feature selection methods could achieve better results if the best terms were picked first from the positive feature set and next, from the negative feature set. Our work differs in that no one-sided metrics were used. However, the weaknesses that our work exposes in two-sided metrics aides in understanding the work that was done to improve MI.

4.4 Word Embeddings

Word embeddings map terms to vectors and are one of the key breakthroughs for NLP. They allow words with similar meanings to have similar representations. This is different from the traditional BOW models, which do

not have a good way of showing that words may be related. Traditional models of word embeddings include continuous bag-of-words, skip-gram, and **Global Vectors for word representations (GloVe)** [28]. GloVe produces term embeddings using a deep neural network based on matrix factorization techniques and is created through a least squares model trained on global word co-occurrence counts [28]. This differs from a skip-gram model, often used to implement Word2Vec, which trains on separate local context windows. For this reason, skip-gram poorly utilizes the statistics of the corpus. Research has shown that GloVe outperforms other traditional models on benchmark tasks, such as the word analogy dataset, as well as on named entity recognition [28].

In 2018, a new word embedding model called **Bidirectional Encoder Representations from Transformers (BERT)** was developed, which achieved state-of-the-art results on 11 natural language programming tasks [13]. Unlike traditional word embedding models, BERT uses an attention mechanism to create context-specific word embeddings [13]. Models such as GloVe use the same embedding for words, even if the context of the word differs [28]. However, while BERT is a more powerful model, it is limited to sequences no longer than 500 tokens [13]. As stated above in Section 3, each document in our dataset has an average length of 3,000 tokens and cannot be easily broken into smaller chunks, because there is no metadata showing which portions of the text correspond to which labels. For this reason, we focus on GloVe, which has achieved state-of-the-art performance among the traditional NLP tasks [7, 28] GloVe provides pre-trained word vectors that can be used with text sequences of any length, because the vectors are static and simply applied to the given texts [28]. The one limitation of using GloVe on long text documents is whether the machine learning algorithm that is trained over the GloVe vectors can handle long sequences, as discussed below.

4.4.1 LSTM Networks. GloVe embeddings are often used as the embedding layer of **long short-term memory (LSTM)** algorithms, a form of recurrent neural network. These are used for text classification tasks, because LSTM networks are able to capture sequential patterns in text that BOW feature sets are not able to capture [7]. It should be noted that while LSTM models can handle long sequences as input, they struggle to remember important past information [38, 39], making the classification of long text documents a challenging problem. Zheng et al. demonstrated that the long term memory of LSTM networks can be improved through the use of an attention mechanism [39]. Attention first became a well-known technique for sequence-to-sequence encoder-decoder models because of the inability of the encoder-decoder model to remember long sequences It is also biased toward weighing more recent information as more important regardless of its relevance [19]. Attention is capable of resolving these issues. For our work, we use self-attention, a form of attention that relates different portions of a sequence to each other to show dependencies between words in the text [19]. This allows the LSTM network to focus on the important sections of the text, regardless of whether they are close to each other in the sequence or not. As an example, self-attention is capable of modeling whether a noun and pronoun are correlated with each other in the text.

5 METHODOLOGY AND RESULTS

This section presents our feature selection and GloVe methodology and results. All methods were tested using fivefold cross validation. Because this is a multi-label dataset, every text document can have multiple independent tactic labels. This is different from other ML problems where a data point can only have one label. To handle this problem, a separate classifier was trained for every label when feature selection was used. Additionally, feature selection was used on the training data of every fold to produce the best set of features given the training data and to not snoop the testing data. For MI and CHI, the selected features were used to train a L. SVM and a **Logistic Regression (L. Reg.)** classifier.

A GloVe embedding was also used with a **bidirectional LSTM (BiLSTM)** architecture. A BiLSTM is composed of two LSTM networks, one going forward through the sequence and one going backwards, which aids in improving context [31]. However, the BiLSTM model we used with a GloVe embedding did not need separate

classifiers and instead outputted a one-hot encoding of 12 neurons, one for each tactic. Only the macro averages scores are reported for the tactics, because we want to capture how well the classifiers perform for each label.

5.1 Feature Selection

The mention of a MITRE ATT&CK tactic may only be a single sentence out of the entire document, which means most of the text is not useful when trying to predict the correct labels. This is one motivating reason to use feature selection methods to help remove the distracting words from the feature set. First, both MI and CHI were applied across the training set to reduce the features. After using feature selection, a TF-IDF approach [10, 17] was used to transform the dataset. Finally, L. SVM, the best-performing method used by Legoy et al., was applied to make predictions on the dataset. We also tested feature selection with L. Reg., another classical machine learning algorithm, to ensure that the chosen features did not just improve one classifier. Other classifiers could be chosen and tested.

5.1.1 MI. MI scores were generated for the training set for each individual tactic label, i.e., the MI scores were first generated for each of the terms in the dataset for each tactic. After scoring each term in the dataset with a MI score, the terms that had the highest MI score were used to set a threshold. This is because MI was scoring all of the most informative words with the same score (the highest score). A lower threshold did not add any valuable features in terms of improving predictions. Thus, any word, with a MI score lower than the terms that had the highest MI score, was removed from the feature set.

The results of this method are shown in Table 2 for method L. SVM+MI and L. Reg.+MI. When MI was applied to L. SVM the precision dropped by 1 percentage point from the baseline L. SVM classifier, and there is a slight improvement in the score of the recall. The same can be seen when MI is applied to L. Reg. The precision drops by close to 1 percentage point from the baseline L. Reg classifier and improves slightly for recall. These results do not indicate any substantial improvement; however, taking note of the weaknesses of MI (described below) indicates areas where the feature selection may be improved upon.

The work by Yang et al. shows that MI is heavily biased toward low-frequency terms, which can create too much noise in the dataset [35]. In addition, the work by Zheng et al., discusses the weakness of using two-sided metrics for feature selection, which tend to place higher relevance on features for the positive class, especially when used with imbalanced datasets [40]. MI, which takes into consideration features for both negative and positive classes, is a two-sided metric. Taking all of this into consideration, low-frequency terms that appear in only one, two, or three documents were removed from the feature set for the positive class. There was no benefit from removing low-frequency terms for the negative class. In fact, it caused a decrease in both precision and recall, so only low-frequency terms for the positive class were removed. After this modification, L. Reg and L. SVM surpassed the scores of the baseline classifiers as seen in Table 2 (rows L. SVM+MI2 and L. Reg.+MI2). For L. Reg (L. Reg.+MI2), the precision score stays almost the same as the baseline L. Reg. classifier, but recall improves by 13.3 percentage points, a substantial increase. L. SVM (L. SVM+MI2) has a precision score of 67.7% and a recall score of 57.4%, beating the L. SVM classifier by 4.9 percentage points and 13.8 percentage points, respectively. The fact that both L. Reg and L. SVM improved so much when this method of feature selection was used further suggests that this methodology is selecting the correct terms.

There are multiple possible explanations for why pulling out low-frequency words from the positive class help. One possible explanation is that MI is biased toward picking more terms from the positive feature set, which caused excess noise. Removing some of those terms reduced this noise and brought more balance between negative and positive features. Second, it can be noted again that the positive labels may only be a single sentence out of an entire document. As such, infrequent terms that appear in those documents are likely to create substantial amounts of noise, which is why the positive class benefits from their removal. In contrast, the negative labels only indicate the lack of the label, and infrequent terms seem to aid in their prediction.

Table 2. Feature Selection Results

Method	Macro Avg Prec.	Macro Avg Recall	Macro Avg $F_{.5}$
L. SVM	0.628	0.436	0.577
L. Reg.	0.668	0.266	0.513
L. SVM+MI	0.617	0.457	0.576
L. Reg.+MI	0.565	0.300	0.480
L. SVM+MI2	**0.677**	**0.574**	**0.654**
L. Reg.+MI2	0.667	0.399	0.588
L. SVM+CHI	0.667	0.414	0.594
L. Reg.+CHI	0.559	0.242	0.443
GloVe Pre-trained	0.412	0.218	0.350
GloVe Weighted	0.379	0.685	0.416
GloVe InfoSec-trained	0.395	**0.698**	0.433
GloVe Attention	0.460	**0.691**	0.493

MI provides the optimal results when considering both precision and recall, as shown in bold for method L. SVM+MI2. While GloVe performs the best on recall as shown in bold for methods GloVe Attention and GloVe InfoSec-trained, it overall is not able to perform well on information security threat reports.

5.1.2 CHI. Next, CHI scores were generated on the training set for each word in the feature set and for each individual tactic label. A p-value of .05, along with one degree of freedom, was used to obtain the critical value of 3.84 from the chi-squared distribution table to ensure significance. Any word with a chi-squared value greater than 3.84 was assumed to be independent of the label and left out of the feature set. The results are shown in Table 2 for rows L. SVM+CHI and L. Reg.+CHI. The CHI metric improved the precision from the baseline (L. SVM) when L. SVM was used, but the average recall score dropped by 2.2 percentage points. When L. Reg was used, CHI does worse on both recall and precision than the baseline L. Reg model. The CHI metric is also a two-sided metric, similar to MI, so it shares the same weakness of prioritizing the positive features. However, removing low-frequency words did not aid the CHI metric. This is likely because CHI is not biased toward the infrequent terms in the same way that MI is.

5.2 GloVe Embedding

We examine a BiLSTM model that is composed of a GloVe embedding layer, a BiLSTM layer, two dense layers, and a dense layer with sigmoid activation that outputs 12 neurons, one for each tactic. In addition, we also test adding a self-attention layer to the BiLSTM model. Binary cross entropy was used for the loss. A threshold of 0.5 was set for the predictions. In addition, the batch size was set to 128, and the model was trained for five epochs when self-attention was not used, based upon trial and error. When self-attention was utilized, the number of training epochs was increased to six. While, the number of training epochs may seem low, using a higher number of epochs quickly led to over-fitting of the model, showing the model's inability to generalize to the information-security data. We examined several GloVe embeddings, two of which are outlined below.

5.2.1 Pre-Trained GloVe. First, we used the common crawl GloVe embedding, created by Stanford [28]. This embedding was trained over 42 billion tokens, with a 300-dimensional vector space, and was used as the embedding layer of the BiLSTM model. Results are shown in Table 2 for the method pre-trained, which shows scores that are much lower than the baseline L. SVM method.

To further aid the BiLSTM model, a weighted binary cross entropy function was created (GloVe Weighted in Table 2). Class weights were first assigned for the positive and negative classes for each tactic. The weighted loss function returns the mean of the weight matrices for each class multiplied together with binary cross

entropy. The average recall is almost 70%, surpassing all methods used so far. The precision, however, dips by 3.3 percentage points from the unweighted BiLSTM model, and the precision is overall much lower than any of the feature selection methods using L. SVM or L. Reg. This indicates that many false positives are returned.

5.2.2 Information-security Trained GloVe. As can be seen from the results using a pre-trained GloVe embedding, the overall precision is low for both methods. This may be due in part to the fact that the dataset is full of highly specific information-security words that the GloVe embedding was never trained over. To test this theory, tools from the GloVe repository were used to create an information-security-specific GloVe embedding. To do so, we downloaded a corpus that contains the first 100 million characters from Wikipedia, which is then supplemented with domain-specific texts, to create a GloVe embedding. Since training a GloVe embedding requires large amounts of data, we had to supplement the information-security documents with those from Wikipedia. In all, a training set of 211,000 information-security documents was collected for training a GloVe embedding. This included 503 malware threat reports downloaded from the Github repository APTnotes [8], around 42,000 entries extracted from Exploit Database [11], around 165,000 CVE entries from MITRE [15], around 1400 cybersecurity blog posts extracted from Krebs on Security [20], and the 1,490 reports from the examined dataset. All of the text documents were tokenized and then used to create a GloVe embedding. The results from using this embedding with the weighted loss function are shown in Table 2. It can be noted that there is a small improvement in results from the weighted LSTM model but precision is still considerably lower from the TF-IDF methods. Last, a self-attention layer, using additive attention, was added to the BiLSTM model to test if self-attention allowed the BiLSTM to focus on more relevant portions of the text. The results can be seen in Table 2. While adding a self-attention layer did improve the overall recall, the results still do not outperform the TF-IDF methods.

5.2.3 Evaluation. Overall, using a BiLSTM with a GloVe embedding scored highest on the overall recall score, with the GloVe embedding trained on the cybersecurity corpus scoring the highest with a recall score of 69.8%. However, precision did not go above 50% for any of the four GloVe embedding methods, although adding a self-attention layer did improve the overall precision score. Overall, MI applied to L. SVM, with low-frequency words pulled out for the positive class, performed the best, with both recall and precision scoring above 50%. It also had the highest $F_{.5}$-score of 65.4%.

While GloVe has often performed better on text classification than TF-IDF methods [7], it seems that in this case the GloVe word embedding did not sufficiently represent a highly domain-specific dataset. While we also tested a GloVe embedding trained over a information-security corpus, it also performed poorly. One possibility is that it was not trained over a large-enough corpus, indicating that many more documents would be necessary to improve the result. In addition, many of these labels might only be represented by a few words out of an entire text document. This is different from other text classification tasks where GloVe has been shown to do well, such as sentiment analysis or classifying the topic of a document [7]. In those scenarios, often most of the text is relevant to the label being predicted. This in part helps explain why feature selection would aid in classification so much, because it removes the noise from the irrelevant portions of text.

Table 3 shows the top 10 contributing words for each tactic for the L. SVM classifier before MI was applied and after MI was applied. This is shown for one fold of fivefold cross validation as an illustration of how the feature set changes. Several things can be noted. First, when L. SVM is used on its own with no feature selection, it can be noted that four of the tactics contain words that refer to specific threat actors or malware types (shown in bold in the table). For instance, the top most contributing word for the Command and Control tactic is "wcry," which refers to the WannaCry ransomware [3]. The tactic Impact is being correlated with three words referring to specific malware variants. This seems to suggest overfitting, where the tactics are being correlated to specific malware variants over actual behaviors. After feature selection is applied, these overly specific malware variants are no longer selected as the top features, allowing for more generalizability of the classifier. However, it should also be noted that when MI is used with L. SVM the Lateral Movement tactic is correlated with the term "duke,"

Table 3. Top Contributing Words

Tactics	L. SVM	L. SVM+MI2
Credential Access	tool keylogger stub ntlm str mimikatz smb hook password credential	controller min document hash conspirator exception keystroke credential password hook
Execution	exe execution backdoor lsa fin guard powershell execute task dde	script job organization http run rule policy task driver command
Impact	ransom recovery **kazuar** flood **shamoon wnry** attack ransomware wiper ddos	adversary system copy hacker campaign packet resource request attack backup
Persistence	run flame kernel path extension shell backdoor hook account persistence	job rights list conspirator agent backdoor load option hook daemon
Privilege Escalation	exe hook detailed fin token shell uac task sid privilege	property daemon shell conspirator express io library task hook privilege
Lateral Movement	wide ursnif flame movement ssh lateral php dcom remote backdoor	**duke** capability pons drive copy atm post erratum machine session
Defense Evasion	evasion control defense backdoor exe bypass gpo payload certificate token	gpo ad issue attribute certificate solution artifact indicator sample control
Exfiltration	conspirator gallmaker dnsname backdoor firewall ftp ddns exfiltration bitsadmin btz	inquiry firewall affair telecommunication name definition sample drive conspirator po
Discovery	footprinting reg finfisher system gather network backdoor trust net information	sample process name trust bookmark consequence like conspirator check address
Collection	**remcos** flame finfisher str implant rat **trickbot** clipboard fin browser	lab domain min function capability format capture sample conspirator keystroke
Command and Control	**wcry** actor beacon port fin implant http dga backdoor communication	port actor proxy paper tl topic conspirator sample communication domain
Initial Access	seg **darkhotel** cpe hxxp removable sqlmap pps fileitem cobalt **emotet**	target organization document device account creator conspirator vulnerability erratum ci

Words in bold refer to specific malware variants or threat actors.

which also refers to a specific threat actor/malware variants [16]. While it is unclear why the terms for this particular tactic seems to decline more overall when MI is used, the other tactics seem to show improvement.

Second, we can see some of the other top words seem to have improved as well. For instance, the top feature for Credential Access when L. SVM is used on its own is "tool" a vague term that could apply to almost anything. When feature selection is applied, the top correlated feature for Credential Access is "controller." MITRE ATT&CK lists "domain controller authentication" as one method of achieving the Credential Access tactic [1]. The same can be said for the Execution tactic, where "exe" could refer to any file and is not generalizable to all operating systems. The term "script" is more generalizable to any executed malicious process. For the Initial Access tactic, "target" and "organization" make more sense as top terms than "seg" and "darkhotel" (darkhotel refers to a cyberattack group [37]). While many of the top terms do seem to show improvement for the tactics when feature selection is

used, we should point out some odd terms that have made their way into the top terms. For instance, the word "like" is listed among the top terms for the Discovery tactic for no obvious reason, which could warrant a deeper look at the texts to understand why.

6 CLASS IMBALANCE METHODS

After selecting MI along with L. SVM as the best-performing method, we did further work to increase the performance of the classifier. As can be noted in Table 1, many of the tactic labels contain class imbalance. This section focuses on different methods of correcting the imbalance in the data, which may be contributing to the classifiers struggling to label samples that contain a tactic correctly. Previously, sampling techniques have often been applied as a method of correcting the imbalance within the data [33]. Oversampling is used to create more samples of the minority class by duplicating the minority class or generating synthetic samples of the minority class; however, oversampling can be prone to over-fitting on the data. In contrast, undersampling removes samples of the majority class until there is an even distribution of samples. The work by Legoy et al. on the same dataset used resampling of the minority class in an attempt to improve results, but their work did not show any substantial improvement for recall or precision. We examine two different methods of correcting data imbalance.

6.1 SMOTE: Synthetic Oversampling

SMOTE is an oversampling method that creates synthetic data samples of the minority class [9]. The synthetic samples are created from the feature space by interpolating between k-nearest neighbors of the same class. For this work, SMOTE was applied to the training set with 3-nearest neighbors after feature selection had been used to pick the optimal set of features. Again, L. SVM was used to make predictions on the test set.

6.2 Snorkel: Weak Supervision

Another method to address data imbalance is to label more data. However, this is often a time-consuming task, because most labeling has to be done by experts who go through the text documents manually. Snorkel is an implementation of weak supervision where users can generate labeled training data by using labeling functions, which apply well-known heuristics and rules to the task [29]. The labeling functions may be overlapping and have noise. Snorkel then uses an ML model to predict the best set of labels for unlabeled text documents [29].

Here 503 unlabeled malware threat reports were downloaded from the Github repository APTnotes [8]. To create the labeling functions, heuristics were needed for each tactic. Snorkel seems to work best when there are labeling functions for both the positive and negative classes. This was a challenging problem given that the negative class represented the absence of a tactic. However, there is the possibility that the absence of a tactic makes other malware tactics and behaviors more likely to show up in the text. Using that assumption, the training data were used to find high-frequency words that mainly show up for the positive label or mainly show up for the negative label. These lists of words were then used to create the labeling functions.

Once the labeling functions were created, the unlabeled set of documents from APTnotes was labeled using the Snorkel model. The Snorkel predictor either predicts one of the labels (positive or negative in this case), or it abstains from giving a prediction when the model is uncertain or there is no relevant labeling function for a text document. In this case, we made the assumption that abstain predictions could also be seen as an indicator of the absence of a label, so all of them were changed to negative labels. Snorkel was applied after using MI on the dataset.

6.3 Evaluation

The results of SMOTE and Snorkel on L. SVM are shown in Table 4. The table shows the results of using SMOTE and Snorkel without any feature selection, as well as the results of using SMOTE and Snorkel after MI has been applied. The L. SVM baseline model is also shown here.

Table 4. Data Balancing Results

Method	Macro Avg Prec.	Macro Avg Recall	Macro Avg $F_{.5}$
L. SVM	0.628	0.436	0.577
L. SVM+MI2	**0.677**	**0.574**	**0.654**
L. SVM+SMOTE	0.604	**0.581**	0.599
L. SVM+Snorkel	**0.663**	0.317	0.544
L. SVM+MI2+SMOTE	0.621	**0.656**	0.628
L. SVM+MI2+Snorkel	**0.667**	0.417	0.596

Snorkel generally improves the precision while SMOTE increases the recall as shown in bold. Mutual information feature selection (MI2) increases both the precision and the recall, as shown in bold for method L. SVM+MI2.

It can be seen that SMOTE without any feature selection greatly improves the average recall score from the baseline classifier, with an improvement of 14.5 percentage points while mostly maintaining precision—dropping by 2 percentage points. SMOTE, with MI, improved even further, gaining 21 percentage points in recall over the baseline classifier, with precision only being 1 percentage point less than the baseline. This affirms that MI is finding a more optimal feature set, which allows SMOTE to create synthetic samples that have far less noise. Additionally, SMOTE with MI outperforms the recall of just using MI by 8 percentage points, although it does worse in terms of precision.

Snorkel, however, only gains 3.5 percentage points in terms of precision against the baseline classifier, and the recall score drops over 10 points to 31.7%. Using MI alongside Snorkel improves the results but still does worse than just using MI on its own. It is likely that Snorkel could be improved upon if better heuristics were given to the labeling functions. The biggest challenge is in creating the labeling functions that indicate the absence of a tactic.

7 DISCUSSION

In summation, the key findings of this research show that many of the state-of-the-art techniques used in NLP need to be adapted specifically for the information security domain. Our methods provide a 6% gain in the $F_{.5}$-score from the research done by Legoy et al. when MI was used as a feature selection tool when predicting tactics, showing an overall improvement in precision. Additionally, a combination of SMOTE and MI saw a gain of 6.5% in total recall, with both methods achieving higher $F_{.5}$-scores. Of note is the fact that MI performed so well in this scenario when the infrequent words were removed for the positive label in the training set. Past research has shown that other feature selection methodologies, such as CHI, have generally out performed MI when used for classification tasks [30, 34]. This work indicates that MI can outperform CHI when it is pruned of some of the infrequent terms that it is biased toward. This work also showed that traditional feature selection out performed GloVe on an information security-specific task, likely because the pre-trained GloVe embeddings have been trained over more generalized text documents.

While the results from this research show improvement from past research, more work is needed to have reliably labeled threat reports that can be used to automate the labeling of malware datasets. In the future, this work could benefit from more research into creating a domain-specific word embedding. This work could also benefit from being extended to the MITRE ATT&CK techniques as well, although generating the feature sets for around 200 technique labels may be a time-consuming task, which is one potential downfall of this approach. However, once generated, the feature sets do not have to be created again unless the training data were to change. Further, it should be understood that the MITRE ATT&CK framework is constantly adapting and combining or adding labels to their system as new behaviors are discovered. While this does not invalidate any of the labels within this dataset as being incorrect, it does give room toward improvement in creating a more dynamic dataset.

Last, while this work can greatly help in labeling datasets of malware, it should be noted that threat reports may not mention every single behavior that a malware sample might actually have.

8 CONCLUSION

Describing the actions taken by malware would be beneficial to cyber analysts. However, most malware detection methods do not describe their actions and threat reports vary in their terminology making comparisons difficult. In this article, we have discussed past research that has been done on labeling malware threat reports with behavioral labels, which indicate that non-domain-specific methods typically used for NLP tasks may not be sufficient for information security tasks. These methods include utilizing different kinds of word embeddings, such as BERT and GloVe. For our task, we chose to use a GloVe embedding due to the length of the threat reports. In addition, we also considered other methods that could improve upon the results of Legoy et al. This led to research in using feature selection methodology as a way of creating optimal feature sets to improve overall precision and recall. In addition we also attempted to balance the dataset with other pre-processing methods such as SMOTE and Snorkel. This work showed that feature selection outperformed GloVe when working with a very domain-specific dataset. Finally, we also showed improvements over baseline results when using SMOTE to to create synthetic data to fix the data imbalance.

As malware continues to find novel ways to evade detection, our purpose is to find new ways in which we can automate the labeling of malware with behaviors. The ability to detect malware by the behaviors it exhibits not only aids in detection but also aids the malware analysts, who now have starting places of behaviors that they can look for. Unfortunately, there are not many datasets of malware that are labeled with these kinds of behaviors. The work from this research aims to help with this task by giving a means to utilize all of the prior information that has already been written about malware to create these kinds of labeled datasets.

REFERENCES

[1] 2019. Credential Access. Retrieved from https://attack.mitre.org/tactics/TA0006/.

[2] Webroot. 2020. *Webroot Threat Report.* Technical Report.

[3] 2021. Preventing WannaCry (WCRY) Ransomware Attacks Using Trend Micro Products. Retrieved from https://success.trendmicro.com/dcx/s/solution/1117391-preventing-wannacry-wcry-ransomware-attacks-using-trend-micro-products?language=en_US&sfdcIFrameOrigin=null.

[4] Fatemeh Amiri, Mohammad Mahdi Rezaei Yousefi, Caro Lucas, Azadeh Shakery, and Nasser Yazdani. 2011. Mutual information-based feature selection for intrusion detection systems. *J. Netw. Comput. Appl.* 34, 4 (2011), 1184–1199. https://doi.org/10.1016/j.jnca.2011.01.002

[5] Benjamin Ampel, Sagar Samtani, Steven Ullman, and Benjamin Ampel. 2021. Linking Common vulnerabilities and exposures to the MITRE ATT&CK framework: A self-distillation approach. *CoRR* abs/2108.01696 (2021). https://arxiv.org/abs/2108.01696.

[6] Yevonnael Andrew, Charles Lim, and Eka Budiarto. 2022. Mapping linux shell commands to MITRE ATT&CK using NLP-based approach. In *Proceedings of the International Conference on Electrical Engineering and Informatics (ICELTICs'22).* 37–42. https://doi.org/10.1109/ICELTICs56128.2022.9932097

[7] James Barry. 2017. Sentiment analysis of online reviews using bag-of-words and LSTM approaches. In *Proceedings of the 25th Irish Conference on Artificial Intelligence and Cognitive Science (CEUR Workshop Proceedings, Vol. 2086)*, John McAuley and Susan McKeever (Eds.). CEUR-WS.org, 272–274.

[8] Kiran Blanda. 2016. Aptnotes. Retrieved from https://github.com/aptnotes.

[9] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* 16 (2002), 321–357.

[10] Gobinda G. Chowdhury. 2010. *Introduction to Modern Information Retrieval.* Facet Publishing, London.

[11] Exploit Database. 2003. Offensive security's Exploit Database Archive. Retrieved from https://www/exploit-db.com/.

[12] Luca Demetrio, Battista Biggio, Giovanni Lagorio, Fabio Roli, and Alessandro Armando. 2019. Explaining vulnerabilities of deep learning to adversarial malware binaries. arXiv:1901.03583. Retrieved from http://arxiv.org/abs/1901.03583.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'19), Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[14] 2021. Micro and Macro Averages for Imbalance Multiclass Classification. Retrieved from https://androidkt.com/micro-macro-averages-for-imbalance-multiclass-classification/.
[15] 2022. allitems.txt.Z. Retrieved from https://cve.mitre.org/.
[16] Noora Hyvarinen. 2015. The Dukes: 7 Years of Russian Cyber-Espionage. Retrieved from https://blog.f-secure.com/the-dukes-7-years-of-russian-cyber-espionage/.
[17] Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *J. Document.* 28, 1 (1972), 11–21.
[18] Faith Karabiber. 2021. Learn Data Science—Tutorials, Books, Courses, and More. Retrieved from https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/.
[19] Alexander Katrompas and Vangelis Metsis. 2022. Enhancing LSTM models with self-attention and stateful training. In *Intelligent Systems and Applications*, Kohei Arai (Ed.). Springer International Publishing, Cham, 217–235.
[20] Brian Krebs. 2009. Krebs on Security. Retrieved from https://krebsonsecurity.com/.
[21] Pradnya Kumbhar and Manisha Mali. 2016. A survey on feature selection techniques and classification algorithms for efficient text classification. *Int. J. Sci. Res.* 5, 5 (2016), 9.
[22] Aditya Kuppa, Lamine Aouad, and Nhien-An Le-Khac. 2021. Linking CVE's to MITRE ATT&CK techniques. In *Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES'21)*. Association for Computing Machinery, New York, NY, Article 21, 12 pages. https://doi.org/10.1145/3465481.3465758
[23] Valentine Legoy, Marco Caselli, Christin Seifert, and Andreas Peter. 2020. Automated retrieval of att&ck tactics and techniques for cyber threat reports. arXiv:2004.14322. Retrieved from https://arxiv.org/abs/2004.14322.
[24] Swee Kiat Lim, Aldrian Obaja Muis, Wei Lu, and Chen Hui Ong. 2017. MalwareTextDB: A database for annotated malware articles. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 1557–1567. https://doi.org/10.18653/v1/P17-1143
[25] Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*. Association for Computational Linguistics, 63–70. https://doi.org/10.3115/1118108.1118117
[26] Infosecurity Magazine. 2019. Cybercrime Costs Global Economy $2.9M per Minute. Retrieved from https://www.infosecurity-magazine.com/news/cybercrime-costs-global-economy/.
[27] Fahmi Salman Nurfikri, Mohamad Syahrul Mubarok, et al. 2018. News topic classification using mutual information and bayesian network. In *Proceedings of the 6th International Conference on Information and Communication Technology (ICoICT'18)*. IEEE, 162–166.
[28] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. Association for Computational Linguistics, 1532–1543. https://doi.org/10.3115/v1/D14-1162
[29] Alexander J. Ratner, Stephen H. Bach, Henry R. Ehrenberg, and Chris Ré. 2017. Snorkel: Fast training set generation for information extraction. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD'17)*. Association for Computing Machinery, New York, NY, 1683–1686. https://doi.org/10.1145/3035918.3056442
[30] Monica Rogati and Yiming Yang. 2002. High-performing feature selection for text classification. In *Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM'02)*. Association for Computing Machinery, New York, NY, 659–661. https://doi.org/10.1145/584792.584911
[31] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. 2019. The performance of LSTM and BiLSTM in forecasting time series. In *Proceedings of the IEEE International Conference on Big Data (Big Data'19)*. 3285–3292. https://doi.org/10.1109/BigData47090.2019.9005997
[32] Michael R. Smith, Nicholas T. Johnson, Joe B. Ingram, Armida J. Carbajal, Bridget I. Haus, Eva Domschot, Ramyaa Ramyaa, Christopher C. Lamb, Stephen J. Verzi, and W. Philip Kegelmeyer. 2020. Mind the gap: On bridging the semantic gap between machine learning and malware analysis. In *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security*. 49–60.
[33] Mike Wasikowski and Xue-wen Chen. 2010. Combating the small sample class imbalance problem using feature selection. *IEEE Trans. Knowl. Data Eng.* 22, 10 (2010), 1388–1400. https://doi.org/10.1109/TKDE.2009.187
[34] Yan Xu, Gareth Jones, Jintao Li, Bin Wang, and Chunming Sun. 2007. A study on mutual information-based feature selection for text categorization. *J. Comput. Inf. Syst.* 3 (03 2007), 6.
[35] Yiming Yang and Jan O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML'97)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 412–420.
[36] Yanfang Ye, Tao Li, Donald Adjeroh, and S. Sitharama Iyengar. 2017. A survey on malware detection using data mining techniques. *ACM Comput. Surv.* 50, 3, Article 41 (Jun. 2017), 40 pages. https://doi.org/10.1145/3073559
[37] Kim Zetter. 2014. DarkHotel: A Sophisticated New Hacking Attack Targets High-Profile Hotel Guests. Retrieved from https://www.wired.com/2014/11/darkhotel-malware/.
[38] Jingyu Zhao, Feiqing Huang, Jia Lv, Yanjie Duan, Zhen Qin, Guodong Li, and Guangjian Tian. 2020. Do RNN and LSTM have long memory? In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 11365–11375.

[39] Wendong Zheng, Putian Zhao, Kai Huang, and Gang Chen. 2021. Understanding the property of long term memory for the LSTM with attention mechanism. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM'21)*. Association for Computing Machinery, New York, NY, 2708–2717. https://doi.org/10.1145/3459637.3482399

[40] Zhaohui Zheng, Xiaoyun Wu, and Rohini Srihari. 2004. Feature selection for text categorization on imbalanced data. *SIGKDD Explor. Newsl.* 6, 1 (Jun. 2004), 80–89. https://doi.org/10.1145/1007730.1007741

Received 30 May 2022; revised 13 February 2023; accepted 23 March 2023