Cyber Threat Detection by Leveraging Contextual and Semantic Knowledge

Vishal Kumar Singh, Padmapriya Mohankumar, and Ashraf Kamal

Abstract In the last two decades, the abundant amount of unstructured data available on the Internet is seen as vulnerable to cyber threats. It significantly impacts on individuals, businesses, governments, etc. Considering this, we present a new study for cyber threat detection via deep learning-based approach. To this end, we devise a model called CASKET to detect cyber threat from the input textual data as vulnerable or non-vulnerable. Our proposed model is novel in the sense that it captures both semantic and contextual knowledge from the input textual data to detect cyber threat. The empirical evaluation of this study is conducted on two benchmark datasets. Our proposed CASKET model performs better on both datasets and shows better *F-score* and *Accuracy* values as compared to the existing works and baseline methods.

Keywords Cyber threat detection · Cyber security · Deep learning · Online social media · Information retrieval

1 Introduction

The Internet has become a vital source to generate huge amount of data from numerous online sources, such as social media platforms, blogs, and e-news. These huge numbers often become a good source for further research [1]. It is seen that text-based classification is performed to deal with the classification of social media information for several problems by linguistics researchers [2]. It includes several research problems like satire detection [3], GAN in protein structure [4], mental health exploration [5], stance detection [6], and hate speech [7]. Although, Internet

V. K. Singh (✉) · P. Mohankumar · A. Kamal
PayPal, Chennai, India
e-mail: vishalksingh@paypal.com

P. Mohankumar
e-mail: pamohankumar@paypal.com

A. Kamal
e-mail: askamal@paypal.com

A. Swaroop et al. (eds.), *Data Processing and Networking*, Lecture Notes in Networks and Systems 1288, https://doi.org/10.1007/978-981-96-3102-5_27

is a vital resource for information propagation, but it also presents many opportunities for cyber threats.

Cyber threat can be defined as any potential malicious attempt to damage or disrupt a computer network, system, or device. These threats can come from various sources, such as hackers, cyber criminals, and insiders within an organisation. They can target personal information, sensitive financial data, intellectual property, etc. It surges several challenges based on cyber threat security risks because of the presence of vast heterogeneous data on the Internet. It has become a vital concern for organisations because the cost of cybercrime can reach up to 0.8% of the global gross domestic product.[1]

1.1 Our Contributions

Detection of cyber threats can benefit several sections of society ranging from individuals, organisations, and infrastructures. Admitting this, we present a new approach to deal with cyber threat detection. It introduces a new deep learning-based model called CASKET (*Contextual and Semantic Knowledge for Cyber Threat Detection*) which mainly captures contextual and semantic information from the input text and that supervises our proposed model to detect cyber-threat-related vulnerable data points/instances. It is employed as a binary class problem which takes pre-processed data as input. Thereafter, it passes to the embedding layer, wherein two different embeddings are used. One embedding is BERT[2] which is used to capture contextual information, whereas another embedding is GloVe[3] and that is responsible to obtain semantic information from the outcome of the input layer. The outcome of these two embeddings are passed to the RNN layer, wherein two BiGRUs receive their corresponding embedding layer outcome and that gives spatial and latent information-related sequences from the input text. Thereafter, two attentions are used in the attention layer which focus on the relevant vulnerable tokens present in the input text. One of the attentions is single-headed in the attention layer which takes the semantic knowledge related BiGRU outcome and another is multi-head attention layer which takes contextual knowledge driven outcome. The outcome of these two attentions in the attention layer are combined together in the shared-attention layer which gives concatenated context vector. Finally, the generated concatenated context vector is forwarded to the dense layer followed by *Sigmoid* which is a non-activation function is employed to classify the input text as either vulnerable (i.e., cyber threat) or non-vulnerable (i.e., non-cyber threat). The main contributions of this study are summarised below:

[1] https://www.csis.org/analysis/economic-impact-cybercrime

[2] https://huggingface.co/docs/transformers/en/model_doc/bert

[3] https://nlp.stanford.edu/projects/glove/

- Introduced a new deep learning-based model called CASKET for computational cyber threat detection on textual data.
- Leverage semantic and contextual enrich knowledge representation through our proposed model to diagnose crucial threat-related presence in the input textual data.
- Conducted experiments on two benchmark datasets for empirical evaluation of the proposed model.
- Performed comparative analysis of our proposed CASKET model with existing studies and baselines methods.

2 Related Work

The section presents the existing studies for cyber threat detection. In [8], authors proposed deep learning models like LSTM and auto-encoder to detect cyber threat. In [9], authors proposed a keyword-based self-learned framework called SONAR to detect cyber security events using real-time Twitter. In [10], authors introduced iACE which is an automated tool for Indicators of Compromise (*aka*, IoC) extraction. It performed knowledge graphs and considered shortest path across nodes in the graph for fact-check purposes. In [11], authors employed architecture based on name entity relation for extracting IoC to generate reports based on cyber security. They considered manual features along with attention mechanisms and BiLSTM. In [12], authors proposed an unsupervised approach cyber threat detection on Twitter. In [13] authors adopted a hybrid attention technique for security vulnerabilities identification. In [14], authors applied BiLSTM and attention layers for vulnerability identification in multi-label setting.

In [15], the authors created pipeline and introduced a tool which applied deep neural networks to process cyber security information from Twitter. In [16], authors introduced a system called CyberTwitter. It utilised cyber security intelligence from Twitter to generate different alerts for threats. In [17], authors highlighted a multi-task approach by combining NLP and cyber threat related intelligence. To this end, they have created a pipeline which considered Twitter streams from users' accounts via a shared neural network-based functionality, wherein cyber-security-related content. In [18], authors introduced a framework for detection cyber threat via Twitter. They applied a data-driven approach for classifying tweets based on cyber security-related events. In [19], authors considered multi-task approach for cyber threat detection on Twitter.

3 Proposed Approach

This section presents the proposed approach for cyber threat detection. It consists of problem description, dataset collection and pre-processing followed by in detail discussion of our proposed CASKET model.

3.1 Problem Description

In this study, we propose a two-class problem for cyber threat detection, wherein a particular textual data is classified as either vulnerable or non-vulnerable. The classified vulnerable instance refers to the cyber threat data points and non-vulnerable instance refers to the non-cyber threat data points.

3.2 Dataset Collection and Pre-processing

In this study, we have considered two benchmark datasets. A short description of datasets is given below:

- **Dionísio et al.** [15]: This dataset follows a multi-task approach and collected tweets from Twitter. It has total number of 11,073 tweets as data points, in which 5118 are vulnerable and 5955 are non-vulnerable data points.
- **Behzadan et al.** [16]: This dataset is based *annotation* and *type*. We have finalised 2380 data points if *annotation* as'threat' and *type* as'vulnerability' for vulnerable data points. For non-vulnerable data points, we have taken 2051 data points if *annotation* as'business or unknown' and *type* as 'General'.

Post-collection, we apply numerous pre-processing steps on both datasets to prepare a cleaner version for empirical evaluation of this study. We have filtered mentions, punctuation, unwanted dots, extra white spaces, question marks, characters, URLs, comma, and exclamations from the collected raw datasets. Finally, we convert the whole data point into the lower-case form.

3.3 Proposed CASKET Model

In this section, we present in detail description of our newly introduced CASKET model. Figure 1 presents the architectural work-flow of the proposed CASKET model.

Input layer: The input layer receives the pre-processed data. Firstly, words present in the pre-processed data are tokenised. Each token assigns an index value and map as

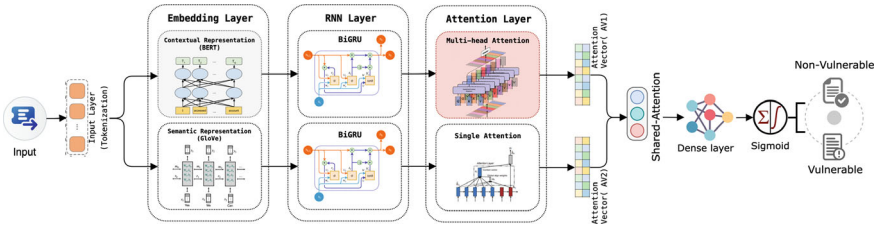

Fig. 1 Architectural work-flow of our proposed CASKET model

a dictionary. As a result, it converts into an input vector, i_v. Thereafter, it is converted into a padded vector through fixed-padding, and accordingly it is transformed into a padded vector, p_v. The main purpose of performing padding mechanism is to maintain the same padding length for all input instances.

Embedding layer: The embedding layer receives the padded vector, p_v generated from the input layer. In this study, we consider two embeddings (BERT and GloVe) which help in generating contextual and semantic knowledge, respectively in our proposed model. BERT, a very famous pre-trained language Transformer model takes input padded vector, p_v to generate relevant and contextual embedded knowledge representation for vulnerable information via 768-dimensional vector. GloVe is another popular embedding used which covers content-based semantic knowledge from the input text. Finally, the encoded representation of both embeddings generates two embedded vectors.

RNN layer: RNN (*aka*, recurrent neural network) is a well-known neural network layer which is designed to handle sequential data from the input. It recognises contextual patterns in sequences for vulnerable or non-vulnerable data via loops within the network to maintain information about previous inputs.

- **BiGRU:** In this study, BiGRU (*aka*, bidirectional gated recurrent unit), a popular two-gated popular kind of RNN which functions in opposite directions. We have taken two BiGRUs, wherein one BiGRU receives the outcome of the embedding vector from BERT to obtain context-based latent semantic vulnerable or threat-oriented feature representation as forward GRU sequences and backward GRU sequences with respect to forward and backward directions, respectively. Equations 1 and 2 present the forward GRU and backward GRU, respectively which receive BERT embedded vector. Equation 3 shows a combined contextual representation of BiGRU which operates parallel and in opposite directions for BERT embedding.

Likewise, another BiGRU in our proposed model takes the embedded vector generated from GloVe. It generates content-based latent semantic vulnerable or threat-oriented feature representation as forward GRU sequences and backward GRU sequences with respect to forward and backward directions, respectively which

receive GloVe embedded vector. Equation 4 shows a combined context-based representation of BiGRU which operates parallel and in opposite directions for GloVe embedding.

$$\overrightarrow{\text{gru}_{\text{f}_{\text{bert}}}} = \overrightarrow{\text{GRU}}\left(b_{\text{t}}, \overrightarrow{h_{\text{t}-1}}\right) \tag{1}$$

$$\overleftarrow{\text{gru}_{\text{b}_{\text{bert}}}} = \overleftarrow{\text{GRU}}\left(b_{\text{t}}, \overleftarrow{h_{\text{t}-1}}\right) \tag{2}$$

$$\text{gru}_{\text{bert}} = \left[\overrightarrow{\text{gru}_{\text{f}_{\text{bert}}}}, \overleftarrow{\text{gru}_{\text{b}_{\text{bert}}}}\right] \tag{3}$$

$$\text{gru}_{\text{glove}} = \left[\overrightarrow{\text{gru}_{\text{f}_{\text{glove}}}}, \overleftarrow{\text{gru}_{\text{b}_{\text{glove}}}}\right] \tag{4}$$

Attention layer: In deep learning, attention layer allows the model to focus on specific parts of the input sequence when generating each part of the output sequence. Considering this, in this study, we consider two attention (multi-head and single-head) layers.

- **Multi-head attention**: It has more than one attention head. The output from each head is concatenated and linearly transformed. It captures the relationships within the input sequences generated from the BERT-based combined BiGRU sequences from Eq. 3. Our proposed model jointly gives threat or vulnerable information based on context from different positions of the input text via multiple heads. As a result, it generates contextual vector, c_{b} as output which is passed to the shared-attention layer.
- **Single-head attention**: It captures the relationships within the input sequences generated from the GloVe-based combined BiGRU sequences from Eq. 4. It provides valuable semantic insights into the dependencies between sequential tokens and emphasises on vulnerable tokens based on semantic information. Consequently, it generates contextual vector, c_g as output which is passed to the shared-attention layer.

Shared-attention layer: This layer gives a combined representation of the outcome generated from the previous layer which leverages both context and semantic knowledge. It takes two contextual vectors generated from the attention layer. It outputs a concatenated contextual vector, c_{v}, as given in Eq. 5 and passed to the next layer.

$$c_{\text{v}} = c_{\text{b}} + c_{\text{g}} \tag{5}$$

Dense layers and Classification: Our proposed model receives the concatenated contextual vector, c_{v}, in the dense layer. Thereafter, Sigmoid (a well-known non-activation function for binary classification problems) performs the task of final classification of the input text as either vulnerable or non-vulnerable.

Table 1 List of deep learning-based parameters settings

Parameters	Values
Dimension (BERT)	768
Dimension (GloVe)	200
Learning rate	le–5
Optimizer	Adam
Epoch (early stopping)	25
Padding sequences	128
Epoch (early stopping)	30

4 Experimental Setup and Results

In this section, we present the experimental settings and results with comparative analysis.

4.1 Experiment Settings

The empirical evaluation of this study is executed using Intel processing machine, NVIDIA GPU, MacOS Sonoma Operating system, and 64-GB RAM. The proposed model is execution on PyTorch[4] (a famous deep learning library in Python). Further, deep learning-based parameter settings used in this study are given in Table 1.

4.2 Results and Comparative Analysis

In this study, datasets are split as 80% for training and 20% for testing for experimental evaluation. Table 2 shows the performance evaluation results on two datasets. It is observed that our CASKET model shows excellent *F-score* and *Accuracy* on both datasets. The proposed model receives the highest *F-score* of 0.95 and *Accuracy* of 0.94 on Dionísio et al. [15] dataset. It has also performed better on Behzadan et al. [16] dataset. It achieves *F-score* of 0.93 and *Accuracy* of 0.92.

Also, observe that CASKET outperforms two previous works and five different combinations of baseline methods. In previous works, Qian et al. [12] receive better results across two datasets. CASKET shows the positive difference of (+ 0.03) and (+ 0.05) for *F-score* and *Accuracy*, respectively as compared to Qian et al. [12] work on Behzadan et al. [16] dataset. In baseline models, the combination of (BERT and BiLSTM) performs better on both datasets. CASKET shows the positive difference

[4] https://pytorch.org/

Table 2 Performance evaluation results on benchmark datasets

Datasets →	Dionísio et al. [15]		Behzadan et al. [16]	
Methods ↓	F-Score	Accuracy	F-Score	Accuracy
Proposed model	0.95	0.94	0.93	0.92
Wu et al. [13]	0.60	0.76	0.81	0.80
Qian et al. [12]	0.55	0.74	0.90	0.87
CNN + BiLSTM	0.81	0.81	0.82	0.83
BERT + CNN	0.85	0.76	0.83	0.79
BERT + BiLSTM	0.88	0.84	0.89	0.85
RoBERTa + CNN	0.79	0.71	0.82	0.76
RoBERTa + BiLSTM	0.85	0.77	0.84	0.76

of (+ 0.04) and (+ 0.07) for *F-score* and *Accuracy*, respectively as compared to the combination of (BERT + BiLSTM) on Behzadan et al. [16] dataset.

Figures 2 and 3 show the train versus test accuracy on two datasets. Observe that, CASKET receives best-fit scenario in terms of train versus test and performance-wise it shows better accuracy value on two datasets.

Although, Dionísio et al. [15] dataset performs marginally better than Behzadan et al. [16]. It shows 0.95 train accuracy and 0.94 test accuracy values for Dionísio et al. [15] dataset, whereas it 0.93 train accuracy and 0.92 test accuracy values for Behzadan et al. [16] dataset.

Also, it can be seen from Figs. 2 and 3 that CASKET outperforms previous works and baseline methods in terms of train versus test accuracy values. Qian et al. [12] perform better in previous studies across both datasets. Our CASKET performs 5.68% and 5.74% better than Qian et al. [12] in terms train and test values, respectively. In different combinations of baseline models, (BERT + BiLSTM) perform better in terms of train versus test accuracy values. Our CASKET performs 8.13% and 8.23% better in terms of train and test values, respectively.

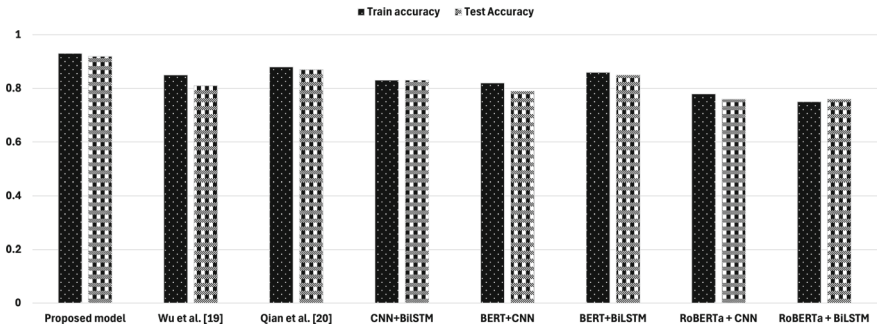

Fig. 2 Train versus test accuracy on Dionísio et al. [15] dataset

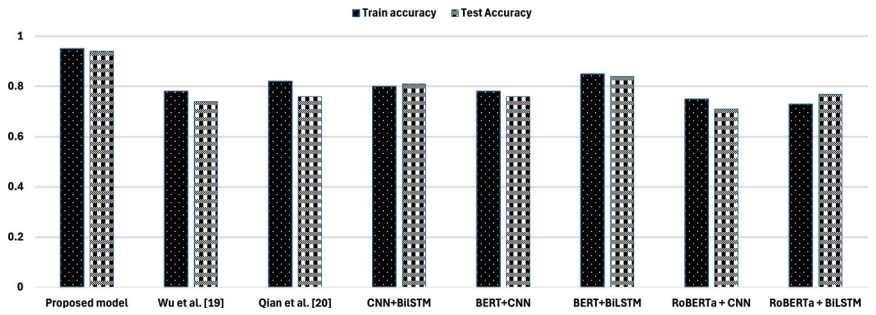

Fig. 3 Train versus test accuracy on Behzadan et al. [16] dataset

Hence, these results highlight that CASKET performs efficiently due to the consideration of the contextual and semantic knowledge representation. Also, inclusion of bidirectional RNN like BiGRU and concatenation of different forms of attention layers significantly contribute in enhancing the model performance.

5 Conclusion and Future Works

This study has presented a new model known as CASKET which leverages deep learning approach to detect cyber threat in textual data. Our model has been designed with the purpose to generate context and semantic knowledge which can be used for cyber threat detection. It has included *input*, *embedding*, *RNN*, *attention*, *shared-attention*, and *output* layers. Our proposed CASKET model has been evaluated on two benchmark datasets. It has achieved the highest *F-score* and *Accuracy* values of 0.95 and 0.94, respectively across both datasets. It has also performed better than the comparable studies and baseline methods. The evaluation of this study on multi-modal settings in future could be an interesting research exploration task. Moreover, considering multi-lingual data to evaluate this study can be one of the worth research directions in future.

References

1. Singh VK, Mohankumar P, Kamal A PayPal Inc, 2024. Digital verification of users based on real-time video stream. U.S. Patent Application 18/059,212
2. Kamal A (2021) A unified data mining approach for detecting figurative language in Twitter https://shodhganga.inflibnet.ac.in/handle/10603/441185
3. Kamal A, Abulaish M, Jahiruddin (2024) Contextualized satire detection in short texts using deep learning techniques. J Web Eng 23(1):27–52
4. Faizi SAA, Singh NK, Kamal A, Raza K (2024) Generative adversarial networks in protein and ligand structure generation: a case study. In: Deep learning applications in translational bioinformatics. Academic Press, Elsevier, pp 231–248

5. Kamal A, Mohankumar P, Singh VK (2022) IMFinE: an integrated BERT-CNN-BiGRU model for mental health detection in financial context on textual data. In: Proceedings of the 19th international conference on natural language processing (ICON). ACL, IIIT Delhi, India, pp 139–148
6. Singh VK, Mohankumar P, Kamal A (2023) Fin-STance: a novel deep learning based multi-task model for detecting financial stance and sentiment. In: 2023 14th international conference on computing communication and networking technologies (ICCCNT). IEEE, IIT Delhi, India, pp 1–6
7. Kamal A, Anwar T, Sejwal VK, Fazil M (2023) BiCapsHate: attention to the linguistic context of hate via bidirectional capsules and hatebase. IEEE Trans Comput Soc Syst (TCSS) 11(2):1781–1792
8. Yazdinejad A, Kazemi M, Parizi RM, Dehghantanha A, Karimipour H (2023) An ensemble deep learning model for cyber threat hunting in industrial internet of things. Digit Commun Netw 9(1):101–110
9. Le Sceller Q, Karbab EB, Debbabi M, Iqbal F (2017) Sonar: automatic detection of cyber security events over the Twitter stream. In: Proceedings of the 12th international conference on availability, reliability and security, pp 1–11
10. Liao X, Yuan K, Wang X, Li Z, Xing L, Beyah R (2016) Acing the ioc game: toward automatic discovery and analysis of open-source cyber threat intelligence. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp 755–766
11. Zhou S, Long Z, Tan L, Guo H (2018) Automatic identification of indicators of compromise using neural-based sequence labelling. In: Proceedings of the 32nd Pacific Asia conference on language, information and computation. ACL, Hongkong, pp 849–857
12. Bose A, Behzadan V, Aguirre C, Hsu WH (2019) A novel approach for detection and ranking of trendy and emerging cyber threat events in Twitter streams. In: Proceedings of the IEEE/ACM international conference on advances in social networks analysis and mining, pp 871–878
13. Wu H, Dong H, He Y, Duan Q (2023) Smart contract vulnerability detection based on hybrid attention mechanism model. Appl Sci 13(2):1–25
14. Qian S, Ning H, He Y, Chen M (2022) Multi-label vulnerability detection of smart contracts based on bi-LSTM and attention mechanism. Electronics 11(19):1–18
15. Dionísio N, Alves F, Ferreira PM, Bessani A (2019) Cyberthreat detection from Twitter using deep neural networks. In: 2019 International joint conference on neural networks. IEEE, pp 1–8
16. Mittal S, Das PK, Mulwad V, Joshi A, Finin T (2016) Cyber Twitter: using Twitter to generate alerts for cybersecurity threats and vulnerabilities. In: ASONAM, pp 860–867
17. Dionísio N, Alves F, Ferreira PM, Bessani A (2020) Towards end-to-end cyberthreat detection from Twitter using multi-task learning. In: IJCNN. IEEE, Glasgow, UK, pp 1–8
18. Behzadan V, Aguirre C, Bose A, Hsu W (2018) Corpus and deep learning classifier for collection of cyber threat indicators in Twitter stream. In: IEEE big data. Seattle, WA, USA, pp 5002–5007
19. Ji T, Zhang X, Self N, Fu K, Lu CT, Ramakrishnan N (2019) Feature driven learning framework for cybersecurity event detection. In: ASONAM, pp 196–203