Keystroke Logging: Integrating Natural Language Processing Technique to Analyze Log Data

Disha H. Parekh, Nehal Adhvaryu, Vishal Dahiya

Abstract**: Cyberwarfare is observed very frequently as always some or the other country is targeting to ruin its enemy country by hacking confidential data from vital computer systems. This has led to dangerous international conflicts. Hence, to avoid illicit entry of other than military person or a government official several tools are being used today as spyware. Keyloggers are one of the prominent tools which are used in today's world to obtain secret or confidential data of a legitimate and contradictory a malicious user too. These keyloggers are advantageous and taken up positively for monitoring employee productivity, for law enforcement and the search for evidence of the crime. While it's negative illegitimate use includes data theft and passwords. The keylogger is today witnessed as a malicious attack and is looked upon as a security threat. But every coin has two sides. Keylogger actually helps in avoiding several security breaches and also aids in detecting several crimes across the net world followed by other fellow countries. This fact has motivated to write this paper and as a consequence, an experimental analysis too was carried out in order to conclude that keyloggers' log file helps identify the person by analyzing proper pattern of the words entered in the file. This paper focuses majorly on the aspect of natural language processing, where a log file obtained thru keylogger software is thoroughly processed via the algorithm as described in the paper. The results yielded a fair understanding of the results obtained as one can easily identify the words used and on the basis of that can also know the type of person on the other end with his ideas, malicious one or of a legal kind.

Keywords* : *Keyloggers, Spyware, Cyberwarfare, Cyberwar

I. INTRODUCTION

Cyberwarfare refers the exploiting of digital attacks such as computer viruses, hacking or intruding and malicious attacks by one country to disrupt the imperative computer systems of another, with the intention of creating harm, decease and demolition. Future wars will see malicious users using computer code to attack an opponent's infrastructure, combating alongside troops using predictable weapons like guns and missiles.

Revised Manuscript Received on January 30, 2020.
* Correspondence Author

Disha H. Parekh*, Department of Computer Science, Indus University, Ahmedabad, Gujarat, India. Email: disha.hparekh213@gmail.com

Nehal Adhvaryu, Assistant Professor, Computer Science, Indus University, Ahmedabad. nehaladhvaryu@gmail.com.

Dr. Vishal Dahiya, Head of Department, Department of Computer Science Sardar Patel University, Vidyanagar, Gujarat India. Email: cs.hod@indusuni.ac.in.

A vague world that is filled with spies, hackers and top clandestine digital arms projects, cyberwarfare as increasingly common and dangerous trait of global conflicts.

Though cyberwarfare usually refers to cyber-attacks committed by one nation on another, it can also portray attacks by terrorist groups or hacker groups intended at furthering the goals of particular nations. Avoiding such forgery attacks and stop cyber stalking, keyloggers work as tool that can help in spying the intruders.

Keylogger is a hardware or software plugin which secretly captures all the keystrokes entered through the keypad of a typing device, without the consent of user. It can affect a desktop or laptop keyboard as well as keypad of smart-devices. The keystrokes get recorded in the form of logs and hence this process is called keylogging, while the tool or the device is called as a keylogger [1].

The logs are stored in the device and then are sent to the receiver via email or some other method as set by the intruder. In fact this type of spying technique can be applied to gain positive and negative, both the outcomes. The choice of using it in either of the way is purely dependent on the user's intention.

There are two types of keyloggers:

Hardware Keyloggers: They are tiny devices fit as an add-on in the computer system to capture and detect the keystrokes. These type of keyloggers are attached in the wifi router, under keyboard or behind the CPU to capture the keystrokes. Nowadays, even optical keyloggers, for wireless devices, are observed that captures keystroke through electromagnetic fields.

Software Keyloggers: Any non-physical technique used for capturing the keystrokes is called a software keylogger and is more destructive than hardware keylogger. These keystroke loggers can be installed in the Operating system, root directory, virtual machines as well as web-forms or any web-scripts [2].

The paper is divided into several sections. The second section consists of papers reviewed to collect the basis of this paper and the basics of keyloggers. Section three consists of methodology that is used in this paper combining keylogger software and python scripts. Fourth section shows the algorithmic evaluation of the stages involved in methodology. Fifth section shows the implementation and the results obtained thru an online tokenizer.

While sixth shows the analysis carried out for the same. Seventh section is very important that shows the results of the experiment carried out than by a coding in python and using Anaconda Navigator for the result analysis and evaluation. Eighth section shows the conclusion of the paper.

II. BACKGROUND

As mentioned in the introduction, a keylogger is a hardware device or software program that records real time activities of a computer user. It can be programmed to store the captured data locally or remotely. It may record all keystrokes or may be sophisticated enough to monitor specific activity-like opening a web browser pointing to your online banking site. Such software can be used maliciously to obtain confidential user information. As such, commercial software versions are often used by parents, spouses or corporations to monitor an unsuspecting user.

There are various types of keyloggers found in present day. All the types of keystroke loggers are divided into major two types as mentioned in introduction. A diagrammatic representation of the types has been shown in the Figure1. The explanation of each type is beyond the objective of this paper.

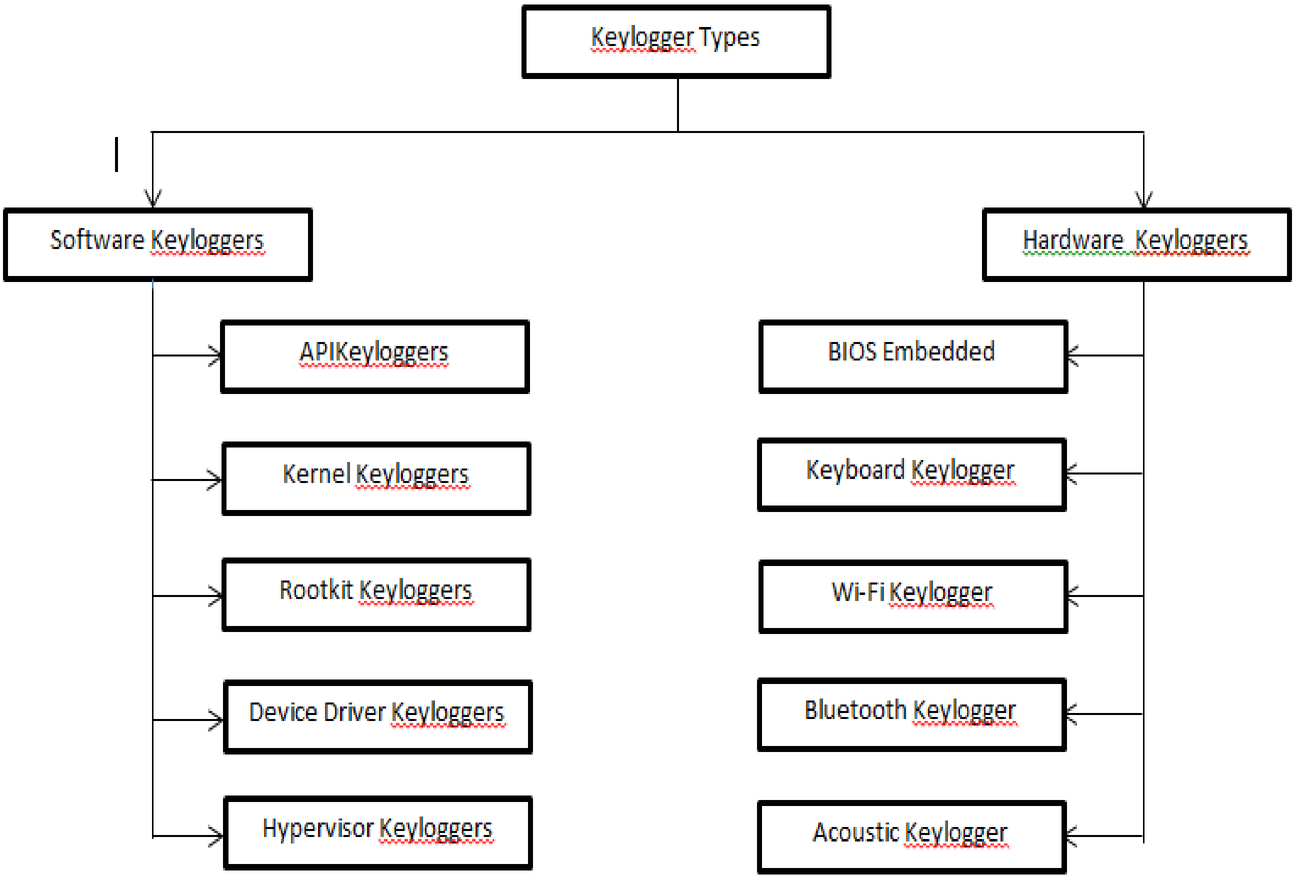


Fig 1: Types of KeyLoggers

Various research studies have examined the current state of keyloggers and how they can play an invaluable role in cyber-security. Some university projects have provided very interesting data:

University of Caen (France): combined keystroke dynamics and 2D face recognition/biometric fusion methods for purposes of identification and authentication [3].

Stanford University: developed a framework called “Telling Human and BOT Apart” a remote biometric system based on keystroke dynamics-designed to fight against spoofing attacks which permits botnets to match a user’s keystroke sequences [4].

While there is a paper on analyzing the Biometric Systems for Authentication which shows various methods of attempts to compromise ones identity and thus allow the user’s authenticity [5].

III. METHODOLOGY

This paper proposes a very novel idea of using keyloggers as a source to identify the person or a user that uses vital computers like government computers, or may be server or some highly confidential system of any organization.

This method is divided into few steps which are carried out sequentially only after a log file is obtained via some software keylogger. Our carried out experiment is shown in the next section.

Here we propose a method to combine keylogger functionality with a recent area of focus today in the world of research, i.e. Data Science. In this area, there is one budding branch under Artificial Intelligence, called as Natural Language Processing, in short described as NLP. It deals with analyzing, understanding and generating the languages that humans use naturally in order to interface with computers in both written and spoken contexts using natural human languages instead of computer languages.

The procedure used after a log file is obtained through keylogger is distributed in steps mentioned in implementation part with its pseudo-code.

IV. ALGORITHM EVALUATION

The implementation part here in the paper mentions the methodology portion. Here, we have shown the pseudo-code in sequential order, which needs to be perform exactly with the same steps.

1. **Obtain a log file:** At the very first instance, a log file from any of the keylogger software is to be obtained. This log file usually consists of details like keystrokes pressed; file name, timestamp and even the changes that a user would have made in the file. In this paper, a Refog Keylogger was used to capture the file written in MS Word.

2. **Tokenization:** Tokenization is a process used in today's most recent technology called Natural Language Processing, in short NLP, to scrutinize word, punctuations, symbols, whitespace in different tokens. It will work in the following manner:

   Input: Unicode string
   Output: Doc object, where Doc object is a sequence of token objects, which will require a special class called Vocab to create Doc object.

   ```
   for token indoc:
    print(token.text, token.pos_, token.tag_, token.is_alpha_, token.lemma, token_punct_, token.is_stop_)
   ```

   Doc Object → encoded to hash values and generates doc object in array of tokens.

3. **Lemmatization:** In the next step, lemmatization is carried out which assigns a correct base form of words tokenized. Here we will require a word net lemmatizer to give a tokenized word a base form lexically. This will be carried out in following manner [6].

   ```
   Import word_net_lemmatizer()
   lemmatizer = word_net_lemmatizer()
   str→ tokenize (keylogger_log_file)
    for word instr
      print(lemmatizer.lemmatize(word))
   ```

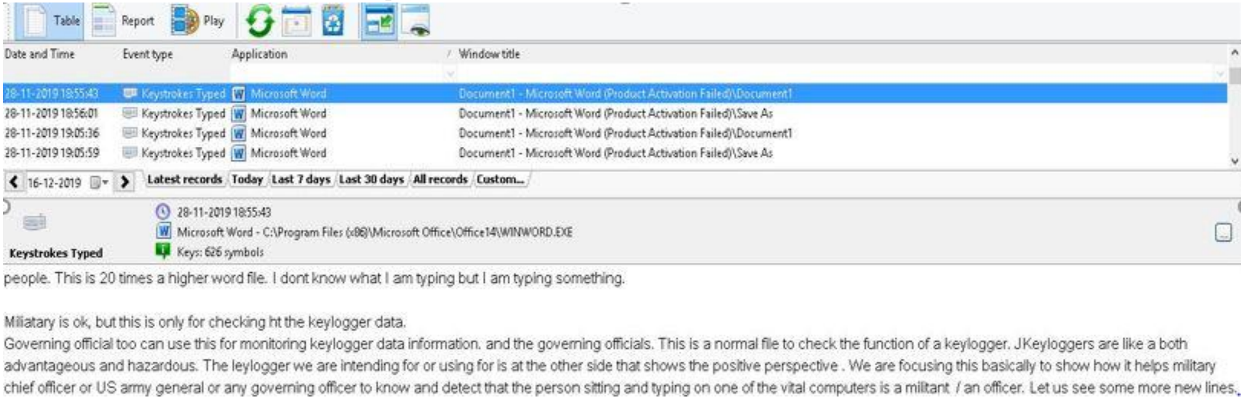
4. **Named Entity Recognition:** This is going to be the last step which will aim in finding the named entities in text and classify it into pre-defined categories like name of person, location, organization, timestamp, title, etc. To carry out this we shall be in requirement to import word tokenizer and a chunk of certain words present in net chunk package. The pseudo-code for it would be:

   ```
   str→tokenize(log_file)
     for str in log_file
       print ne_chunk (word_tokenize (str))
   ```

This method will help in scrutinizing the log file followed by tokenizing the sentence in words, punctuations, timestamp and symbols. After that assignment of lexically analyzed grammatical word to the tokenized word will be carried out which at last will be generating a named entity record with different headings like name of person and his id and the timestamp and even the organization name and title. This may then be fed into some analyzer script which is a trained script consisting of some keywords that we shall match the final data with. After that if we come across any positive false results, we shall be able to identify the person with name and source.

V. IMPLEMENTATION

The keylogger used here for our experiment is refog free keylogger. The sample file was written in MS document and this file was made to record in the refog keylogger. A file was made in MS Word and the keystrokes were captured in the keystroke logger. The output generated in the refog personal monitor keylogger is shown in figure 2.



Date and Time	Event type	Application	Window title
28-11-2019 18:55:43	Keystrokes Typed	Microsoft Word	Document1 - Microsoft Word (Product Activation Failed)\Document1
28-11-2019 18:56:01	Keystrokes Typed	Microsoft Word	Document1 - Microsoft Word (Product Activation Failed)\Save As
28-11-2019 19:05:36	Keystrokes Typed	Microsoft Word	Document1 - Microsoft Word (Product Activation Failed)\Document1
28-11-2019 19:05:59	Keystrokes Typed	Microsoft Word	Document1 - Microsoft Word (Product Activation Failed)\Save As

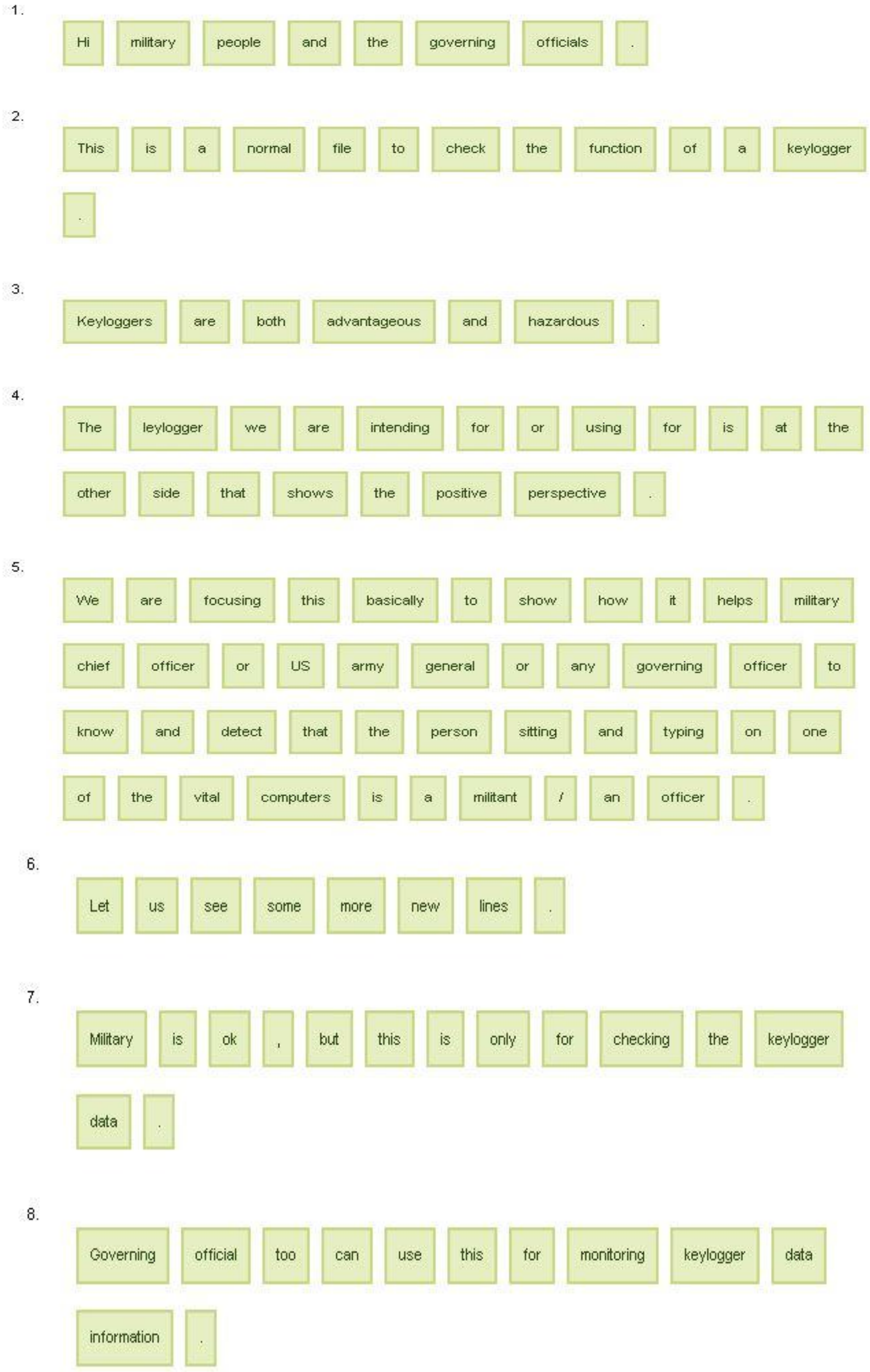
Keystrokes Typed
28-11-2019 18:55:43
Microsoft Word - C:\Program Files (x86)\Microsoft Office\Office14\WINWORD.EXE
Keys: 626 symbols

people. This is 20 times a higher word file. I dont know what I am typing but I am typing something.

Miliatary is ok, but this is only for checking ht the keylogger data.
Governing official too can use this for monitoring keylogger data information. and the governing officials. This is a normal file to check the function of a keylogger. JKeyloggers are like a both advantageous and hazardous. The leylogger we are intending for or using for is at the other side that shows the positive perspective . We are focusing this basically to show how it helps military chief officer or US army general or any governing officer to know and detect that the person sitting and typing on one of the vital computers is a militant / an officer. Let us see some more new lines.

Fig 2: Refog Personal Monitor Keylogger Log File

Next the tokenization was carried out with the help of a tokenizer tool Python NLTK tokenizer [7]. The output generated was of different forms. Out of which we are going to process only pattern based tokenization which is shown in figure 3. This NLTK tokenizer is widely available online and is free of cost. Though this file was not actually helpful for further analysis carried out in section VII of experimental results, but it has been shown for the students or researchers learning the basics of the tokenization.



```
1.
Hi | military | people | and | the | governing | officials | .

2.
This | is | a | normal | file | to | check | the | function | of | a | keylogger
.

3.
Keyloggers | are | both | advantageous | and | hazardous | .

4.
The | leylogger | we | are | intending | for | or | using | for | is | at | the
other | side | that | shows | the | positive | perspective | .

5.
We | are | focusing | this | basically | to | show | how | it | helps | military
chief | officer | or | US | army | general | or | any | governing | officer | to
know | and | detect | that | the | person | sitting | and | typing | on | one
of | the | vital | computers | is | a | militant | / | an | officer | .

6.
Let | us | see | some | more | new | lines | .

7.
Military | is | ok | , | but | this | is | only | for | checking | the | keylogger
data | .

8.
Governing | official | too | can | use | this | for | monitoring | keylogger | data
information | .
```

Fig 3: File with tokenized data in a particular pattern

Tokenization has been done online and for experimental results this was not sufficient, hence a python code was written and implemented to analyze the log file and tokenize it further. Secondly, this file was then made to process for lemmatization and stemming of data. This work can further be extended to also evaluate and perform sentimental analysis with NLP code in python. This will be performed in the upcoming paper, to notice what kind of person is sitting on the computer and also will let us analyze the sentiments or the user's idea with the processing of log file further. Thus it is one of the innovative approaches created by us in order to use keylogger log file with positive aspect.

VI. ANALYSIS

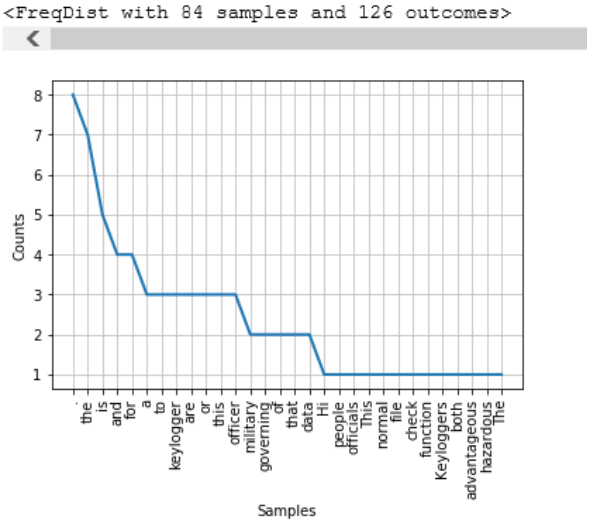
The above mentioned pseudo-code was implemented till tokenization.

This data showed that the text file we wrote was tokenized in form of pattern where whitespace also was used and the punctuations were also tokenized. But this tool was unable to tokenize the data in form of lexical grouping, hence based on the analysis done; a python script for tokenization and lemmatization needs to be developed to tokenize words lexically including the grammar rules as well as dictionary words.

VII. EXPERIMENTAL RESULTS

The normal text file as shown in figure 2 was used to experiment the procedures of tokenization, stemming and lemmatization. For the said experiment, we used Anaconda, Jupyter Notebook and a code for the above stages was done in Python.

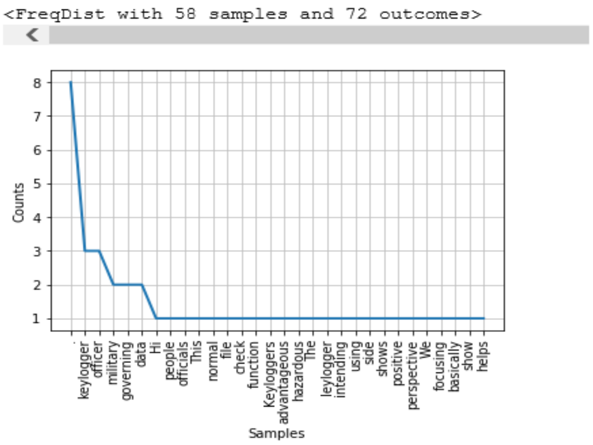
The result was obtained using NLTK package, where first the tokenization was carried out, which was done in just a 0.01 µs. The data tokenized was then made to check the frequency of the word occurrence, which was plotted using matplotlib. The graph obtained is indicated below:



```
<FreqDist with 84 samples and 126 outcomes>
```

Fig 4: Graph showing the occurrences of word after tokenization

After obtaining the frequency of word occurrence, text filtration was carried out, where stop words like ‘for’, ‘to’, ‘if’, ‘from’, were removed and the result obtained is depicted in the below graph:



```
<FreqDist with 58 samples and 72 outcomes>
```

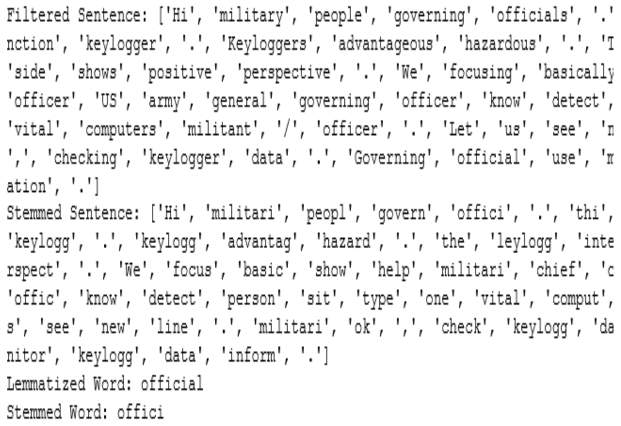
Fig 5: Graph showing word occurrences after removal of stop words

Upon comparing the two graphs it is evitable that words like ‘the’, ‘is’, ‘and’, ‘or’, ‘a’ are removed and the result obtained is as below:

Table – 1: Comparative results of tokenized data to that of filtered data

Frequency Distribution of word	Tokenized Data	Filtered Data	Resultant Data (Words Eliminated)
Samples	84	58	**26**
Outcomes	126	72	**54**

After obtaining the filtered data from tokenized words, stemming and lemmatization was carried out. As an example here, official word was stemmed to office word as indicated in below figure:



```
Filtered Sentence: ['Hi', 'military', 'people', 'governing', 'officials', '.'
nction', 'keylogger', '.', 'Keyloggers', 'advantageous', 'hazardous', '.', 'T
'side', 'shows', 'positive', 'perspective', '.', 'We', 'focusing', 'basically
'officer', 'US', 'army', 'general', 'governing', 'officer', 'know', 'detect',
'vital', 'computers', 'militant', '/', 'officer', '.', 'Let', 'us', 'see', 'n
',', 'checking', 'keylogger', 'data', '.', 'Governing', 'official', 'use', 'm
ation', '.']
Stemmed Sentence: ['Hi', 'militari', 'peopl', 'govern', 'offici', '.', 'thi',
'keylogg', '.', 'keylogg', 'advantag', 'hazard', '.', 'the', 'leylogg', 'inte
rspect', '.', 'We', 'focus', 'basic', 'show', 'help', 'militari', 'chief', 'c
'offic', 'know', 'detect', 'person', 'sit', 'type', 'one', 'vital', 'comput',
s', 'see', 'new', 'line', '.', 'militari', 'ok', ',', 'check', 'keylogg', 'da
nitor', 'keylogg', 'data', 'inform', '.']
Lemmatized Word: official
Stemmed Word: offici
```

Fig 6: Example of stemming from a lemmatized word

Lastly after the stemming and lemmatization, a code for tagging of words with part of speech was carried out as a result of which the obtained result was all the filtered data was tagged with POS. The chief goal of Part-of-Speech POS) tagging is to classify the grammatical group of a given word to a noun, pronoun, adverb, adjective, etc. based on the context. POS tagging looks for associations within the sentence and allocates a matching tag to the word.

Thus these tokenized and filtered data after proper stemming and allocating POS tag to tokens, we can enhance the code to know the sentiments of the user at the other end. Researchers interested can conclude their work with text classification and carry out sentiment analysis of any input text that has been obtained with the help of keyloggers.

VIII. CONCLUSION

This paper demonstrates a novel idea of using log file obtained via keylogger and then analyzing this file with the recent artificial intelligence based technique of natural language processing. In this paper, for experimental analysis, the text file mentioning few words like “government”, “official”, “military”, “militant” and “governing” were captured through keylogger known as Refog keylogger. This file was further tokenized and lemmatized to analyze it further for the type of user. This work is a novel approach to show the advantages of keylogger as a tool that supports to know the facts and also helps in providing higher security.

The uniqueness to the paper is implementing this keylog file with today's very much buzzing technology called Artificial Intelligence thru which the paper has proposed an idea of using Natural Language Processing. This was helpful to indicate the ratio of tokenized data for sample and the output with respect to the filtered data after performing stemming and filtration on the file, which is also aid further for reducing the CPU utilization time and thus overall reducing the processing time.

This further will even aid in evaluating the sentiments of a person on the other side and also learn the pattern of his communication by analyzing his data and the words used over the system. This research is a narrative idea which, as per our knowledge, has not been shown till date in the field of data science that takes up keylogger with positive impact.

REFERENCES

1. R. K. R. Venkatesh, "User Activity Monitoring Using Keylogger.," Asia Journal of Information Technology, vol. 15, no. 23, pp. 4758-4762., 2015.
2. P. T. Sahu, " System Monitoring and Security Using Keylogger.," International Journal of Computer Science and Mobile Computing, vol. 2, no. 3, pp. 106-111, 2013.
3. D. R. John Deluca, "A System-Wide Keystroke Biometric System.," Proceedings of Student-Faculty Research Day, CSIS, Pace University , 2011.
4. L. Nystrom, "https://vtnews.vt.edu/articles/2010/11/111110-engineering-yao.html.," https://vtnews.vt.edu/articles/2010/11/111110-engineering-yao.html, March 2011.
5. J. V. MdLiakat Ali, "Keystroke Biometric Systems for User Authentication.," Springer, 2016.
6. N. L. JoëlPlisson, "A Rule based Approach to Word Lemmatization.," Proceedings of IS-2004, researchgate.net, 2004.
7. M. R. Dr. S.Vijayaran, "Text mining: open source tokenization tools – an analysis.," Advanced Computational Intelligence: An International Journal (ACII), vol. 3, no. 1, 2016.
8. S. R. MayukhRath, "Semi Supervised NLP Based Classification of Malware Documents.," International Conference on Information Systems Security, Springer , 2017.

AUTHORS PROFILE

Prof. Disha Harshadbhai Parekh, is working as an Assistant Professor with Indus University, Ahmedabad. She is passionate for research and carries an extensive ability to perform extraordinarily in research aspects. She is currently motivating her students to write papers during their UG level and also helping several PG students for writing quality papers in the field of computer science. She has completed her MCA and M.Phil in Computer Science. Her area of interest lies with cloud computing, security issues with cloud and cyber crimes, data science and NLP approaches. She is in the education field since 2009 and she has a good databank of research papers under several headings, which can be fetched from scholar and can be contacted on disha.hparekh213@gmail.com .

Prof. Nehal Adhvaryu, is an assistant professor with 10+ years of experience in the field of academics in computer science. She is interested in research areas like cloud computing, natural language processing and deep learning. She has done her MCA. She has been a constant motivator for students in developing technical projects. She is bearing very strong technical skills over Java and Python. She has affection towards coding and can be approached at nehaladhvaryu@gmail.com.

Dr. Vishal Dahiya, is working as a HoD of Computer Science Department with Indus University. She has wide experience of 18+ Years. She has completed her Ph.D. in computer science from Sardar Patel University and currently is guiding more than 10 research scholars. She is acting as a chair person of research committee of Indus University for Computer Science and Engineering Departments. She is a motivator and a mentor for students and faculties interested in research. Her research area focuses on Image Processing and Big Data. She has been widely renowned for her literally work on image processing. She can be approached at cs.hod@indusuni.ac.in.