**PAPER-4**

**SMS Phishing and Mitigation Approaches**

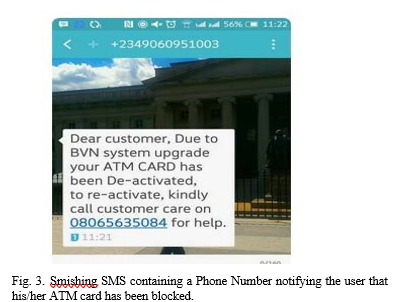
**Author:** Sandhya Mishra, Devpriya Soni

**Year:** IEEE 2019

Introduction:

Phishing and smishing are fraudulent attacks aiming to steal user information via emails, texts, or SMS. Smishing, coined by David Rayhawk in 2006, involves sending text messages with links or contact information to deceive users into disclosing personal or financial details. Attackers often pose as trustworthy entities, such as banks, offering fake prizes or claiming account issues to lure victims into clicking links or providing sensitive information. These messages typically prompt users to visit phishing websites resembling legitimate ones, where they're tricked into revealing login credentials, enabling attackers to conduct unauthorized transactions.



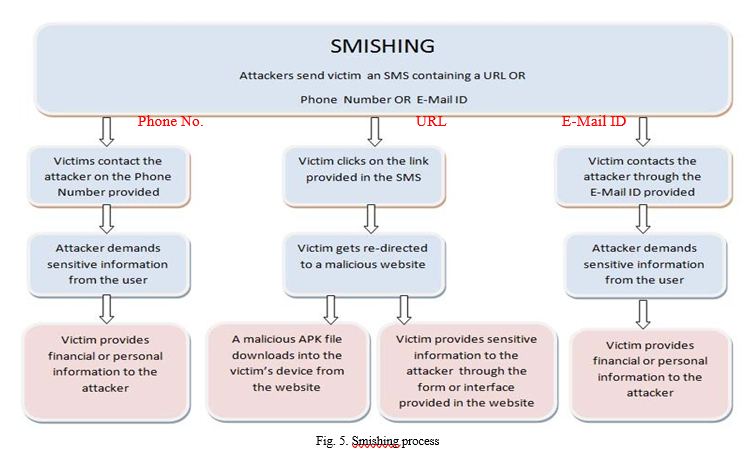


**SMISHING MITIGATION APPROACHES:**

* Downloading Authorized Applications
* Directly Accessing Website
* Using Safe Browsers

**Techniques used to detect Smishing**

* Content-Based Filtering
* Blacklisting
* Whitelisting



**Anti-Smishing solutions:**

* S-Detector
* Rule-Based Framework for Detection of smishing messages in Mobile Environment
* Smishing Classifier
* A Content-Based Approach for detecting Smishing in Mobile Environment
* Security Considerations for SmartPhone Smishing Attacks

**Conclusion:**

Mobile users' lack of security awareness and risky behavior make smartphones prime targets for smishing attacks. Smishers send SMS with malicious links, exploiting the limited display size, leading users to click without recognizing the threat. Users' tendency to download apps without considering their potential risks further exacerbates the issue. This paper explores various smishing attacks, outlines preventative measures, and evaluates techniques for identifying and mitigating smishing threats, providing valuable insights for future research and user education**.**

**PAPER-5**

**Smishing Detector: A security model to detect smishing through SMS content analysis and URL behavior analysis**

**Author**: Sandhya Mishra, Devpriya Soni

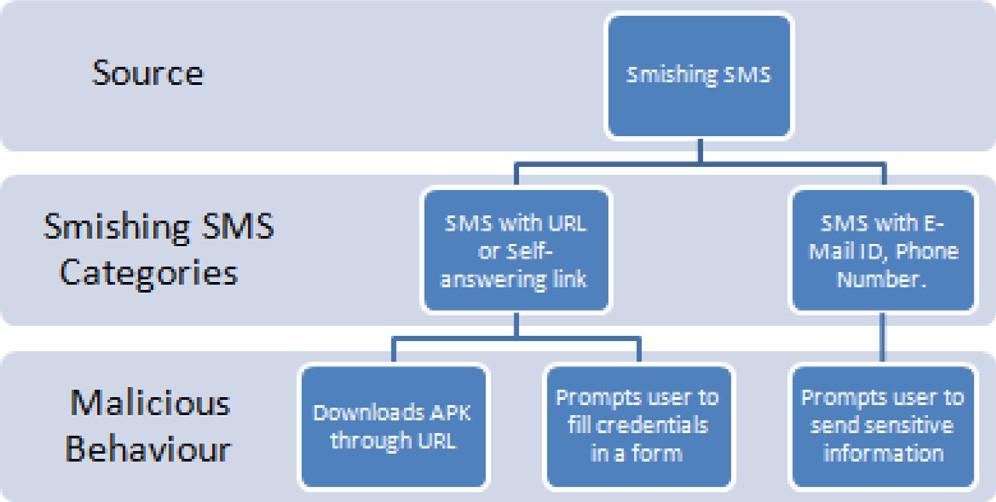
From: ELSEVIER 2020

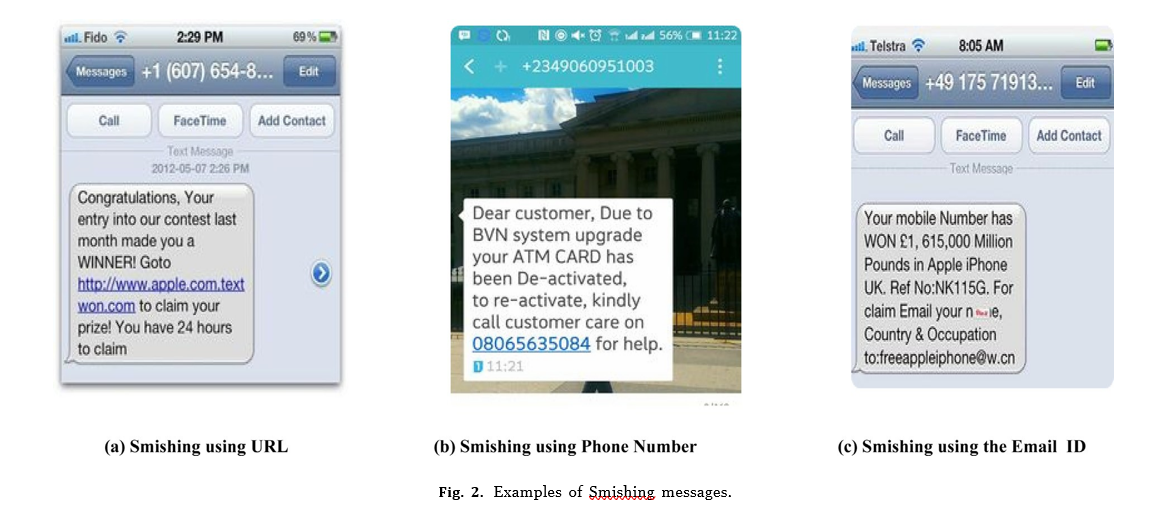
**Abstract:**

Smartphones' popularity and internet connectivity make them susceptible to phishing and smishing. Our 'Smishing Detector' model minimizes false positives in identifying malicious messages, using modules like SMS Content Analyzer, URL Filter, Source Code Analyzer, and APK Download Detector. Experimental results show a strong 96.29% accuracy, outperforming other security models. This system provides effective protection against mobile phishing and smishing attacks.

**Introduction:**

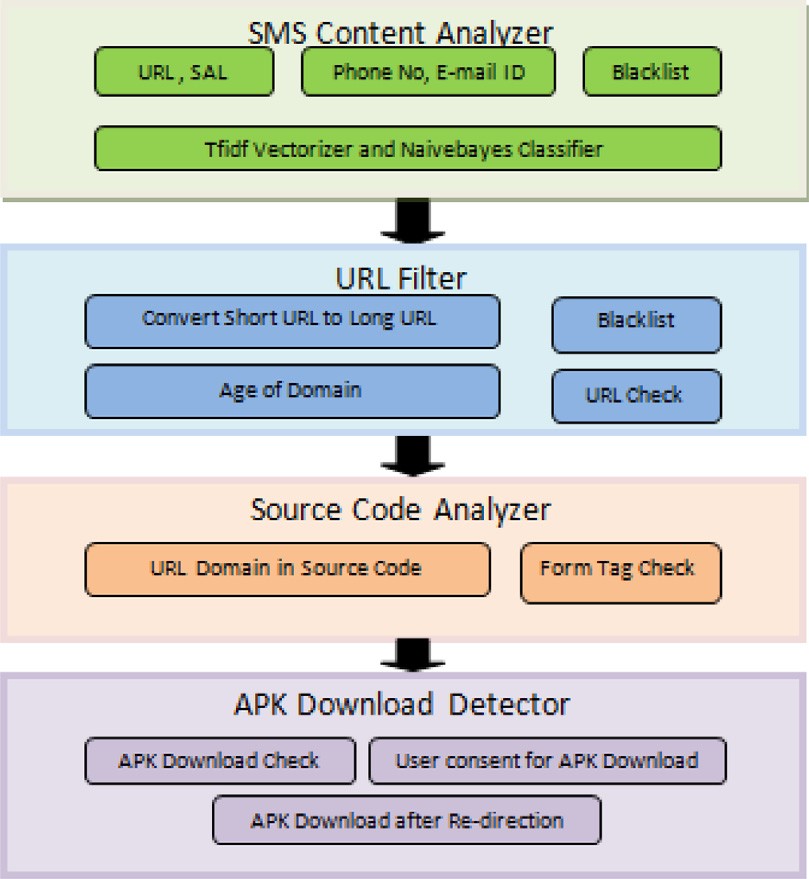
During smishing, attackers design the user interface wisely so that the user is unable to identify the minor differences between a legitimate website and fake website created by the attacker. They copy the source code of the legitimate website to create a fake web page that will look similar to a legitimate web page but these websites could re-direct the user to other malicious links. Also, attackers modify the URL to create a fake URL that looks like URL of a legitimate website.



****

**Proposed work:**

SMS Content Analyzer verifies the presence of URL, self an- swering link (SAL), phone number and email id in the SMS. Messages containing URL or SAL are transferred to URL Filter. Messages containing email id and phone number are processed for blacklist check. Then messages are forwarded for text pre- processing.

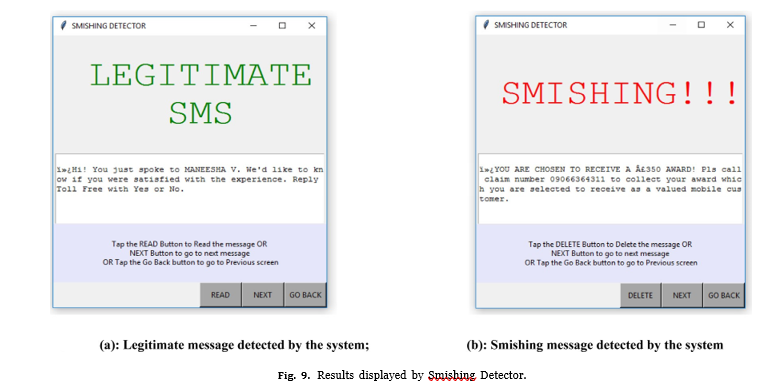


* Source code analyzer
* APK Download Detector
* URL filter

**Implementation result and evaluation:**

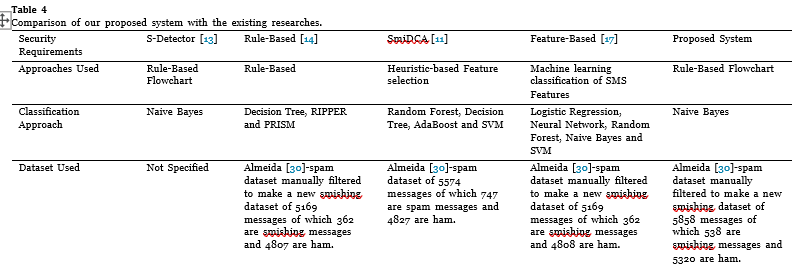
* A smishing detection system, built in Python with Jupyter Notebook, tackles fraudulent messages aiming to steal user data.
* It utilizes four modules:
* SMS Content Analyzer: Examines message content for suspicious keywords and phrases linked to smishing.
* URL Filter: Checks URLs against blacklists of known malicious sites.
* Source Code Analyzer: Scans source code of attached APK files for hidden threats.
* APK Download Detector: Downloads and analyzes APK files linked in messages.
* The system leverages a custom dataset of 5858 messages, including 538 smishing attempts and 5320 legitimate messages.
* Evaluation reveals an impressive 98.5% accuracy in identifying smishing messages.
* This multi-pronged approach combats diverse smishing tactics, including URL manipulation and malicious file downloads.
* Beyond keyword analysis, the system delves deeper into URLs, source code, and APK files for comprehensive detection**.**



****

**Comparative analysis:**

Goel et al.'s Smishing Classifier lacks false-positive elimination, categorizing legitimate APK downloads as smishing. Further verification of website legitimacy is crucial, as login pages and self-answering links from trusted sources may trigger false positives. Our proposed system includes user consent verification during APK downloads, a novel approach to mitigate false positives and enhance detection accuracy in mobile security**.**

****

**Conclusion and Future work:**

The paper presents Smishing Detector, using SMS content analysis and URL inspection to classify smishing messages. It includes modules like SMS Content Analyzer, URL Filter, Source Code Analyzer, and APK Download Detector, achieving 96.29% accuracy with Naive Bayes. Compared to existing models, it offers heightened security against smishing attacks. Future plans involve integrating additional security measures like a Malware detector to enhance application security and prevent personal information leakage. Practical implementation on the Android platform is envisioned, offering real-time smishing detection and prevention.

**PAPER-6**

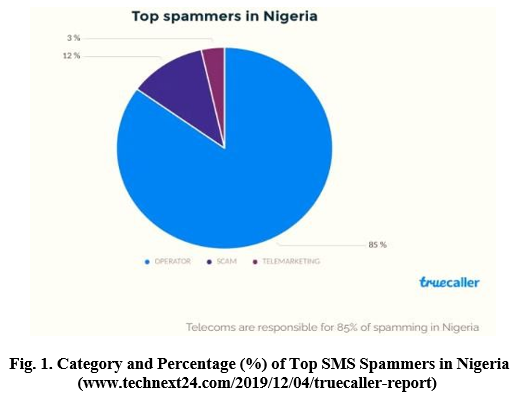
**SMS Spam Detection and Classification to Combat Abuse in** Telephone **Networks Using Natural Language Processing**

**Author:** Dare Azeez Oyeyemi a and Adebola K. Ojo

**Journal & Year**: Journal of Advances in Mathematics and Computer Science2023

Introduction:

The widespread use of mobile phones, particularly for SMS communication, has become integral to modern life, contributing significantly to the GNI of developing countries. However, the ubiquity of SMS has also led to an increase in unwanted spam messages, particularly in countries like Nigeria. Traditional keyword filters and newer techniques like Statistical Learning Theory and ANNs have been used for spam detection, but no single method is fool proof. Applying text mining techniques to SMS can enhance spam detection and classification, reducing network abuse.



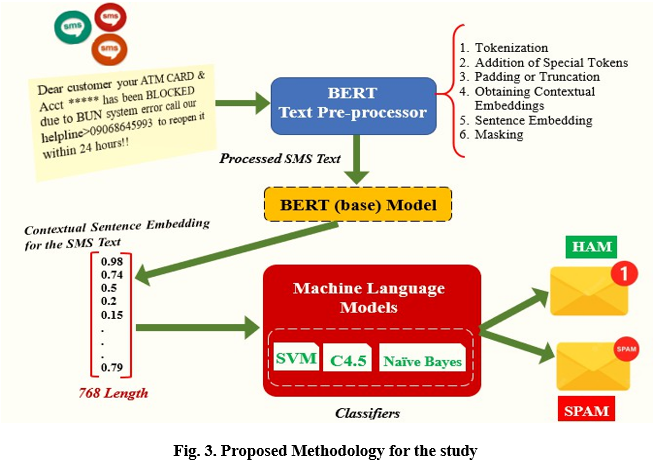
**Research Objective**:

* Pre-process dataset with BERT NLP.
* Extract features using Document Frequency Matrix and BERT.
* Vectorize processed data.
* Evaluate machine learning models: Naive Bayes, Random Forest, Gradient Boosting, Logistic Regression, SVM.

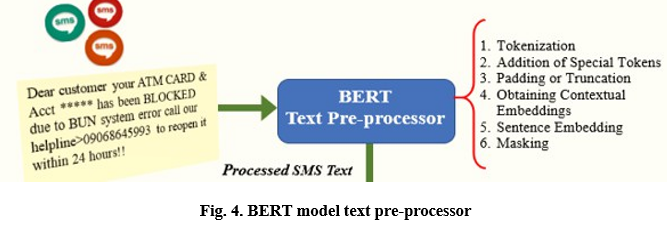
**Scope and Limitations of Study:**

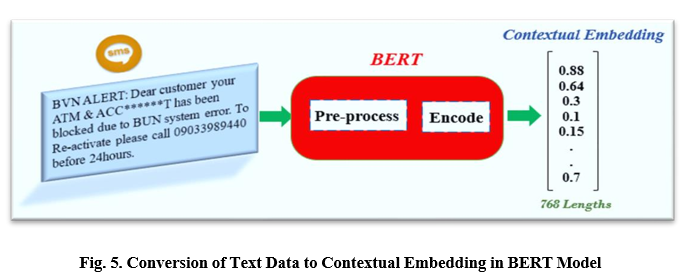
Developing SMS spam detection with machine learning, utilizing diverse datasets and preprocessing for feature extraction. Employing fine-tuned BERT model and traditional ML algorithms for classification, addressing class imbalance through down-sampling. Challenges include limited training data due to downsampling and potential language limitations.

**Methodology:**

****

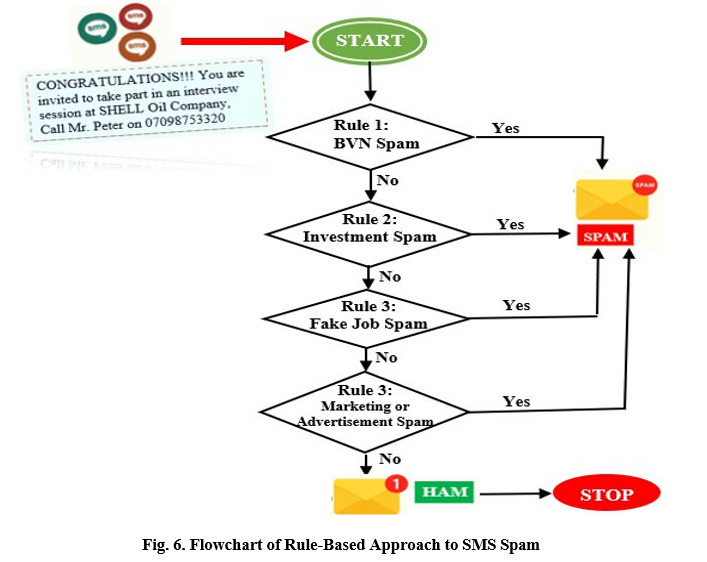
* Cross sectional study
* Data collection phase
* Data cleaning and preprocessing



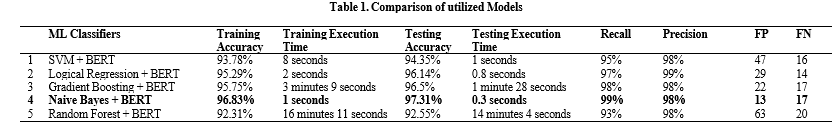


**Results and Discussion:**

* Rule-based filtering approach



* Vectorization
* Comparison between existing models



**Conclusion:**

Study designs ML model to detect spam SMS in Nigeria, comparing it against existing models. Hybrid Naïve Bayes and BERT model achieves 97.3% accuracy, 0.3s execution time. Future directions include testing on new data, incorporating localized datasets, and extending to non-English languages for improved detection**.**

**PAPER-7**

**SMS Spam Detection Using Machine Learning**

**Author**: Suparna DasGupta1, Soumyabrata Saha2\*, Suman Kumar Das3

From: Journal of Physics: Conference Series 2021

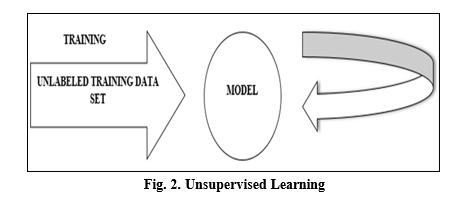
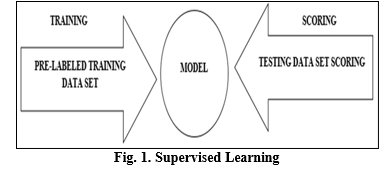
**Abstract:**

SMS, a crucial form of communication, lacks the need for an internet connection but faces vulnerability to hackers and spammers. Authors combat this with a TF-IDF Vectorizer-based system, distinguishing between malicious and non-malicious messages. This dictionary-driven approach aids in classifying SMS as spam or ham, enhancing mobile device security.

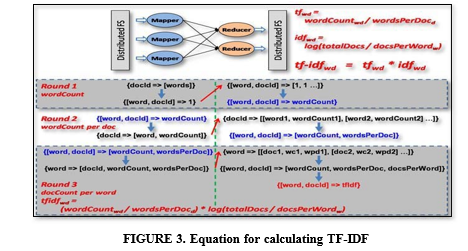
**Introduction:**

SMS, a prevalent mode of communication, lacks the need for internet, making it susceptible to hackers. Authors address this with a Machine Learning system distinguishing spam (SPAM) from non-spam (HAM) messages based on content. Overview includes various ML techniques employed for model development, crucial for mitigating security risks in SMS communication.

* Classification
* Regression
* Supervised learning
* Unsupervised learning



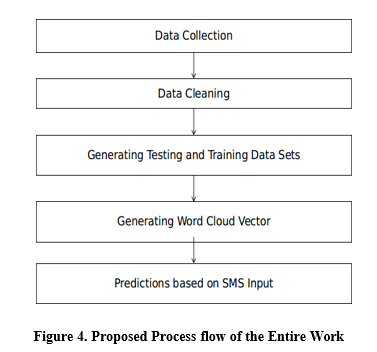
TF-IDF, a weighting factor in ML and text mining, emphasizes rare words' importance while devaluing common ones. It aids relevance ranking and removing stop words, enhancing model accuracy by focusing on meaningful terms. Figure 3 illustrates the TF-IDF formula's mathematical representation



**Methodology:**

**Workflow:**

Authors collected a dataset from Kaggle for experimentation, cleaning it meticulously by removing white spaces, standardizing letter cases, and tokenizing messages. They partitioned the data into training and testing sets, utilizing TF-IDF vectorization to generate word cloud vectors for spam classification. Finally, their model accurately predicted spam or ham messages, enhancing SMS filtering efficiency and user security**.**

****

**Experimentation:**

As a part of experimentation, authors after creating the vector set, passed 2 inputs to test whether or not the model (including the word vectors) is able to check whether the message is SPAM or HAM.

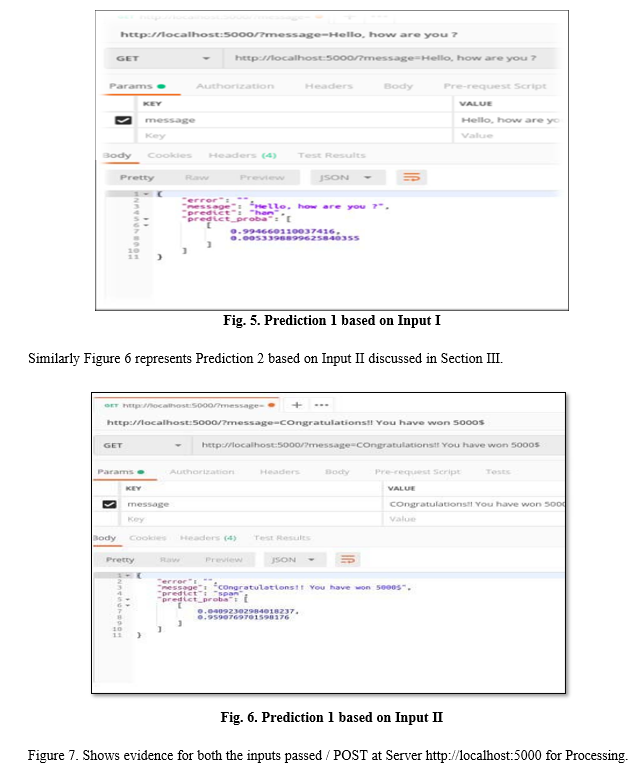
Input I: given to the developed system: “Hello, how are you?”

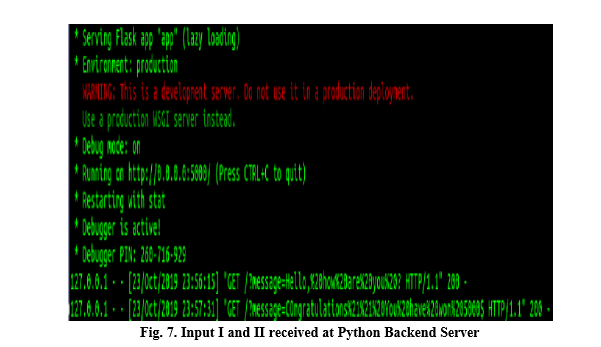
Input II: given to the developed system: “Congratulations!!! You have won 5000$”

Output of the above inputs are discussed in the Section IV of this literature

**Results and Discussion:**

The research utilized a Python-based Flask platform, relying on modules such as BeautifulSoup4, NumPy, scikit-learn, SciPy, Pandas, and Flask. The system output, discussed in Section III, is illustrated in Figure 5. For instance, Input I "Hello, how are you?" was processed by the system.





**Conclusion:**

From the above discussion and experimentation authors have concluded that Machine Learning algorithms can play a vital role in identifying SPAM SMS. The accuracy obtained in this work is more than 95% in both the cases.