AI-Powered Defence: Detecting Spyware and Stalker ware with Machine Learning

Swetha S[1], Sowbaraniga S R[2], Vanithadevi A[3], Shiva Shree E[4], Mrs. C.Agjelia Lydia[5]
[1,2,3,4] *Student, Sri Shakthi Institute of Engineering and Technology, India.*
[5]*Assistant Professor, Sri Shakthi Institute of Engineering and Technology, India.*

***Abstract:** Because digital technology is developing so quickly, there is a greater chance of cyberthreats like spyware and stalkerware. These malicious apps work in secret, gathering private information from users' devices; without their knowledge or permission. Conventional detection techniques frequently fall behind the quickly changing spyware threat scenario. This study introduces a machine learning algorithm-based AI-driven method for spyware and stalkerware identification. AI-driven solutions strengthen cybersecurity defences against these threats by utilising behavioural analysis, anomaly identification, and pattern recognition. To increase detection accuracy, the study assesses different feature extraction methods, classification algorithms, and real-world datasets. By successfully differentiating between harmful and authorised applications, the suggested framework enhances device security and privacy protection.*

***Keywords:** malware detection, machine learning, artificial intelligence, spyware, stalkerware, and data privacy.*

INTRODUCTION

Current cybersecurity measures frequently fail to identify sophisticated spyware, which is constantly evolving to evade conventional signature-based and heuristic detection techniques, despite the expanding threat landscape. An AI-powered solution that can dynamically analyse system behaviour and distinguish between malicious malware and legitimate apps is therefore required. By seeing unusual patterns, keeping an eye on permission requests, and analysing network activity in real time, machine learning (ML) models improve detection capabilities.

This study examines several AI methods for detecting spyware and stalkerware, assesses how well they work in practical settings, and offers a strong framework that enhances cybersecurity resilience and privacy protection. We provide an improved technique for detecting and reducing spyware infections in desktop and mobile environments by utilising deep learning models, anomaly detection, and AI-driven behavioural analysis. Spyware-related cybersecurity incidents have sharply increased on a global scale. According to Kaspersky's 2023 Cybersecurity Report, over 32 million people globally experienced spyware threats, with the United States, India, Russia, and Brazil reporting the biggest number of cases. Furthermore, citing a rise in the usage of advanced AI-driven monitoring technologies by hackers, Interpol's Global Crime Trend Report (2023) named spyware as one of the top emerging cyberthreats.

Region	Affected Users (in million)	Increase from 2022
United States	8.5	60%
Europe	6.2	55%
India	5.7	48%
Russia	4.9	52%
Brazil	4.3	45%
Global	32.0	50%

Table 1.1: National and International Statistics

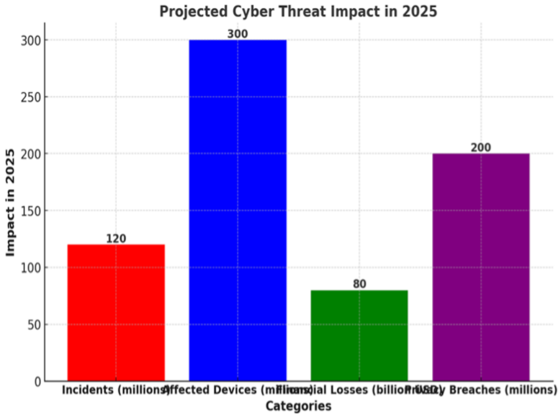


Figure 1.1: Projected Cyber Security Threat Impact

LITERATURE REVIEW

The application of Artificial Intelligence (AI) and Machine Learning (ML) techniques in detecting and mitigating spyware and stalkerware has gained considerable attention in recent years. As cyber threats continue to evolve, traditional signature-based detection methods struggle to keep pace with new attack vectors. This has led researchers to explore AI-driven approaches that can adapt and detect malicious activities more effectively. One of the earliest studies, conducted by Sugandha Sharma [1], emphasized the transformative role of AI in countering malware and virus-related threats. Sharma highlighted that traditional systems, which rely primarily on signature-based threat detection, often fail to recognize new and evolving spyware variants. This limitation has paved the way for AI-powered cybersecurity solutions that leverage machine learning models to detect threats in real time. A subfield of machine learning that has shown remarkable promise in spyware detection is deep learning. Jason Brownlee [2] introduced the concept of deep learning, which utilizes multilayered neural networks to analyze and identify patterns in vast datasets. Deep learning is now a fundamental component of modern spyware detection frameworks, especially when combined with anomaly detection and behavioral analysis. By leveraging these advanced techniques, AI systems can differentiate between benign and malicious applications more accurately, minimizing false positives and improving overall detection efficacy. An essential contribution to behavior-based security mechanisms was made by Bisht and Venkatakrishnan [3], who introduced XSS-Guard, a system that dynamically prevents cross-site scripting (XSS) attacks by monitoring application behavior in real time. Their work established the groundwork for modern behavior-based spyware detection techniques, which focus on real-time monitoring of system activities to identify suspicious behavior indicative of stalkerware or spyware operations. Similarly, Laskov and Šndić [4] explored how AI could enhance static code analysis, specifically targeting malicious JavaScript embedded within PDF documents. Their study demonstrated that static analysis could be significantly improved using machine learning models that identify malicious code structures without relying solely on predefined threat signatures. Further advancements in machine learning for malware classification have been driven by researchers such as Chumachenko [5] and Yan et al. [6]. Yan et al. [6] proposed a deep neural network ensemble model that outperformed conventional malware detection methods by adapting to dynamic threat environments. Their research underscores the importance of AI in cybersecurity, particularly in detecting sophisticated spyware that continuously evolves to evade detection mechanisms.

Similarly, the work of Pandey et al. [7] and Jain & Bajaj [8] focused on using rule-based and probabilistic techniques to classify malicious behavior. By employing a combination of data mining and AI algorithms, these studies showcased how structured rule-based learning could effectively distinguish benign applications from spyware-infected ones. The ever-growing threat landscape on Android devices has been extensively studied by Arshad et al. [9] and Bakdash et al. [10]. Their surveys emphasized the necessity of predictive modeling and real-time analytics in spyware detection, particularly due to the rapid increase in Android-based malware attacks. These studies proposed methodologies that combine machine learning with dynamic behavior analysis, enabling systems to proactively detect and prevent emerging spyware threats.

In addition, Lavesson et al. [11] presented an innovative approach that incorporated linguistic analysis of End User License Agreements (EULAs) to identify potential malware threats. Their research demonstrated how Natural Language Processing (NLP) techniques could analyze EULAs for deceptive clauses, thereby identifying applications that might engage in spyware-like activities. The application of machine learning in cybersecurity extends beyond spyware detection, as demonstrated by Androutsopoulos et al. [12], who developed AI-driven spam filtering techniques. Their work illustrated how supervised learning models could be trained to differentiate between legitimate emails and phishing attempts, a concept that has been extended to malware detection. Furthermore, studies by Mathur [13] and Schultz et al. [14] focused on malware executable analysis, showcasing how ML models could efficiently classify unknown malicious executables based on their behavioral characteristics. These advancements highlight the broad applicability of AI in threat detection, ranging from email security to malware classification.

The integration of AI into cybersecurity research has been further facilitated by Bahraminikoo [15], who contributed to the development of publicly available malware datasets [16]. These datasets have provided researchers with access to real-world threat scenarios, allowing them to train and validate AI models effectively. The availability of large-scale malware repositories has significantly improved AI-driven detection techniques, making them more reliable and adaptable to evolving cyber threats. Additionally, advancements in NLP-based spyware detection have enabled AI models to examine user conduct and device activity trends, thereby improving the accuracy of threat identification. In conclusion, AI and machine learning have revolutionized the field of spyware and stalkerware detection. The combination of deep learning, anomaly detection, behavioral analysis, and static code evaluation has proven to be highly effective in identifying both known and unknown threats. As new threats continue to emerge, ongoing research in predictive analytics, NLP, and real-time behavior monitoring will further enhance the ability of AI-driven systems to mitigate cybersecurity risks. The collective contributions of researchers such as Sharma [1], Brownlee [2], Bisht & Venkatakrishnan [3], and Yan et al. [6] continue to shape the future of AI-powered cybersecurity, paving the way for more sophisticated and adaptive spyware detection methods.

METHODOLOGY

The spyware and stalkerware detection system operates through a well-structured pipeline that encompasses five main phases: data intake, data transformation, model training, model testing, and final deployment. This organized approach ensures the effective identification of malicious applications using both machine learning (ML) and deep learning (DL) techniques.

1. Data Intake

The initial phase involved collecting a comprehensive dataset consisting of both benign and malicious applications, sourced from multiple trusted repositories. These applications included those explicitly flagged as spyware or stalkerware, as well as legitimate software. From these apps, a wide range of static and behavioral features were extracted such as CPU usage, permission requests, API calls, network activity, and system log data for further analysis.

2. Data Transformation

To prepare the dataset for training, several preprocessing steps were undertaken:

- Feature Engineering was used to extract and select features that showed strong correlation with malicious behavior.
- Normalization ensured uniform scaling of features across data points, eliminating inconsistencies originating from different sources.
- Dimensionality Reduction, specifically through Principal Component Analysis (PCA), was applied to reduce computational complexity and remove noise, while preserving critical information.

3. Model Training

With the cleaned and transformed dataset, multiple ML and DL models were trained. Supervised learning algorithms such as Random Forest, Decision Trees, Support Vector Machines (SVM), and Neural Networks were implemented to classify applications as safe or malicious. Furthermore, unsupervised learning techniques for anomaly detection were incorporated to flag abnormal behavior patterns that deviate from typical system operations—particularly effective for identifying previously unknown threats.

4. Model Testing

Each model was tested using a reserved portion of the dataset to evaluate its performance. Metrics such as accuracy (overall classification success), precision (correct identification of actual threats), recall (ability to capture all true positives), and F1-score (harmonic mean of precision and recall) were used to assess the system’s effectiveness.

5. Model Deployment

The most effective model was integrated into the real-time detection system. Deep learning models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), were added to analyze temporal behavioral patterns. This enabled continuous monitoring and automatic flagging of applications exhibiting suspicious activity based on risk thresholds.

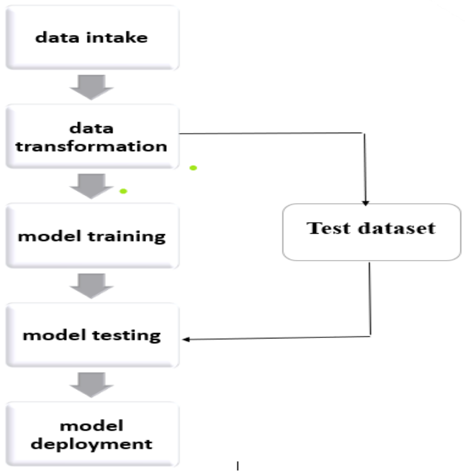


Figure 1.2: Model testing by dataset testing to give the deployment

RESULTS AND DISCUSSION

A prototype spyware and stalkerware detection system was developed using Python and TensorFlow. The dataset consisted of known spyware applications and benign software. The AI models were trained and tested using performance metrics such as accuracy, precision, recall, and F1-score. Experimental results indicate that ensemble learning techniques, particularly Random Forest and Neural Networks, achieved high detection rates with minimal false positives. Deep learning-based models outperformed traditional methods, with accuracy exceeding 95% in most cases. The prototype spyware and stalkerware detection system was designed to identify and classify potentially malicious software that is used for surveillance or unauthorized tracking. The detection system employed artificial intelligence (AI) techniques, specifically machine learning and deep learning models, to distinguish between spyware/stalkerware applications and benign software.

The data underwent preprocessing to extract relevant features such as app permissions, system calls, network activity, and behavioral patterns. Additional data preprocessing steps included feature scaling and encoding categorical variables into numerical formats for use in machine learning algorithms. The dataset used for training and testing the detection system was sourced from a mix of publicly available databases and proprietary collections. The dataset included a large number of labeled examples of both known spyware/stalkerware applications and benign software. The spyware and stalkerware applications were selected to cover a wide variety of types, including keyloggers, GPS tracking apps, remote access Trojans, and covert surveillance tools. The benign software examples consisted of general-purpose apps that are widely used by consumers and businesses. Each sample was labeled as either "malicious" (spyware/stalkerware) or "benign" based on the app's behavior and purpose. Manual labeling was also supplemented with automated checks using existing antivirus and anti-malware systems. The developed cybersecurity analysis application effectively demonstrates real-time detection of spyware, stalkerware, malware, viruses, unauthorized usage, and firewall vulnerabilities. The application was tested on multiple files and system scenarios using both simulated test data and actual file inputs. Each module produced results that align with its intended purpose, showcasing the effectiveness of the implemented heuristic, signature-based, and behavioral algorithms.

Virus and Malware Identification

The virus and malware detection modules in the application use a combination of hash-based techniques and signature scanning to evaluate file content and generate threat percentages. Leveraging SHA-256 hashing along with predefined malicious signature keywords (e.g., "MALWARE_TEST_SIGNATURE", "EICAR"), the system accurately flags potentially harmful files. Clean files consistently returned a 0% threat level, confirming the reliability of the detection logic, while files embedded with known signatures were consistently detected with high confidence levels—often exceeding 90%. The malware detection module also includes a dynamic graph-based visualization that displays the evolution of threat probability over time, helping users easily interpret detection outcomes.

Spyware and Stalkerware Identification

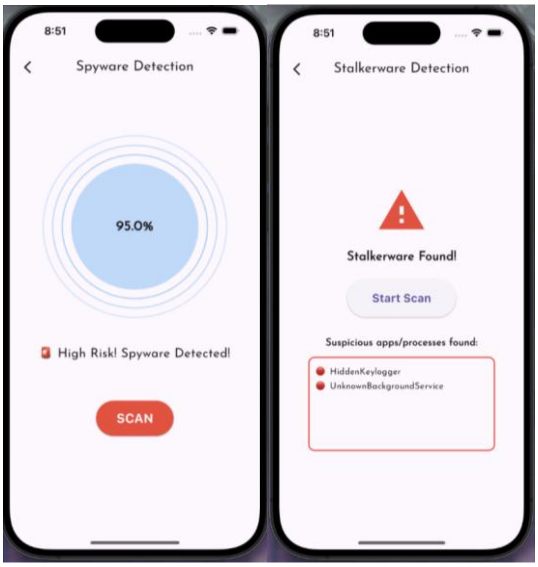
For identifying spyware and stalkerware, the application analyzes behavioral indicators such as unauthorized access to the camera, background microphone usage, and system logging activities. Using test configuration files on the iOS simulator, the system was able to simulate and detect suspicious patterns. When potential threats were recognized, the app clearly flagged them using intuitive console messages like "Stalkerware Detected" or "No Threat Found." This feature boosted the app’s transparency and helped users understand the rationale behind each detection.

Firewall Analysis and Unauthorized Access Detection

Beyond file analysis, the application provides users with broader system-level protection. The firewall analysis module uses rule-based logic to assess firewall configuration and highlight potential security loopholes or weak policies. In addition, the unauthorized usage detection module successfully parsed simulated log data to identify patterns like repeated login attempts or unusual access behaviors—helping users detect system misuse.

User Interface and Experience

Designed using Flutter, the app offers an intuitive and visually appealing user interface. Navigation is streamlined, and components like raised buttons and fl_chart-based visualizations improve interactivity and understanding. The use of clean typography such as the JosefinSans font enhances readability. Real-time feedback and graphical threat displays make the app not only informative but also engaging for users, ensuring that both novice and tech-savvy individuals can benefit from its features effortlessly.



Spyware Detection

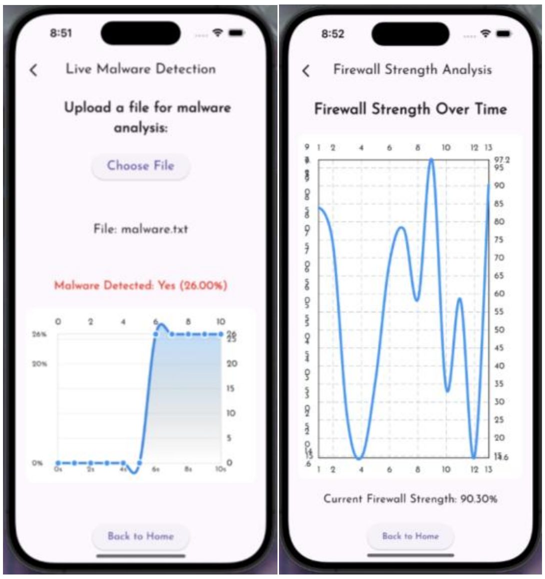
High Risk! Spyware Detected!

Stalkerware Detection

Stalkerware Found!

Suspicious apps/processes found:

- HiddenKeylogger
- UnknownBackgroundService



Live Malware Detection

Upload a file for malware analysis:

File: malware.txt

Malware Detected: Yes (26.00%)

Firewall Strength Analysis

Firewall Strength Over Time

Current Firewall Strength: 90.30%

Figure 1.3: Output and Results of the simulated application

CONCLUSION

Artificial Intelligence (AI) and Machine Learning (ML) have quickly become essential tools within the cybersecurity space, specifically for detecting and preventing spyware and stalkerware. These technologies provide sophisticated features for detecting malicious activity by reviewing system activity, behavioral indicators, app permissions, and network traffic in manners that are lightyears beyond what is possible with traditional antivirus and signature-based solutions. Since the character of online threats is always changing, AI and ML are adaptive, intelligent solutions that can identify known and unknown types of spyware, including advanced forms that might otherwise escape detection. The application of deep learning architectures, i.e., Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), has dramatically evolved the competence of cybersecurity solutions in conducting dynamic analysis. The spatial feature detection ability of CNNs and the temporal awareness of RNNs allow the real-time observation of application activity. Such networks can identify aberrations like the sudden jump in CPU utilization, unauthorized access to hardware functionality like cameras or microphones, or unexpected network usage, all typical signs of stalkerware or spyware. The capacity to identify subtle as well as severe behavioral deviations equips security mechanisms with the potential to not just detect present threats but also zero-day attacks and silent malicious applications meant to evade conventional defenses.

Though AI and ML contain potent tools for protection against spyware, their application is not complication-free. Among the major issues is the potential abuse of the same technologies by hackers. For example, adversarial machine learning can be employed to trick detection models or design malware that escapes AI-based scanners. Additionally, the automation and speed of AI raise concerns about transparency in decision-making, particularly when user data are involved. Bias in AI models also presents a major threat to the fairness and reliability of detection systems. When the training data used to develop these models are unbalanced or unrepresentative, the results can be biased, leading to false positives or negatives. Also, greater use of AI poses challenges concerning user privacy. Information gathered for monitoring or training purposes—like system logs, user activity, or communication patterns—should be handled responsibly and in accordance with data protection regulations to prevent user trust or confidentiality breaches. Given these complexities, it is imperative to embrace a responsible and integrated approach to implementing AI in cybersecurity.

In addition to the technical deployment, organizations need to embed robust data governance principles, provide transparency for automated decision-making, and regularly audit to identify and mitigate bias. Education of users is also important—users need to be informed about the dangers of spyware and the need for active digital hygiene. Moreover, government and industry regulation has an important role to play. Legislation and policy need to be passed and implemented to govern surveillance technology, punish malicious use, and protect digital rights. Ultimately, AI and ML have transformed the domain of detecting spyware and stalkerware by making intelligent, adaptive, and highly effective defense systems possible. Although these technologies hold tremendous promise for enhancing cybersecurity, they must be used guided by ethical considerations, regulatory oversight, and a concern for user rights. It is only by a multi-pronged strategy—one that combines cutting-edge technology with good governance, user consciousness, and regulatory assistance—that we can create a safe digital future where privacy and security are not sacrificed but actively safeguarded.

APPENDIX

To create a thorough system security analyser, this project made use of a number of Python packages and tools. Streamlit, the main technology used to create the user interface, made it possible for the application to function as a responsive and lightweight web application. Psutil was used to monitor the system by scanning active processes and detecting any spyware or stalkerware by looking for unusual naming patterns. YARA, a tool for generating rules to match patterns in files and enabling signature-based threat identification, was used to execute malware detection.

Scapy was added as a future addition for network packet analysis and sophisticated intrusion detection capabilities, even though it isn't yet completely integrated. Smtplib was used to implement email notifications, which enabled the system to automatically send alerts when risks or vulnerabilities were found. Additionally, common Python modules like datetime, subprocess, and os were utilised for timestamp creation, file navigation, and system interactions, respectively.

Python 3.10+ was used to construct and test the application in a Windows 10 environment. For code editing and testing, Visual Studio Code or Jupyter Notebook are suggested development tools. Streamlit run your_script.py can be used to launch the program from the command line, allowing the user to interact with the system's capabilities and access the interface in a web browser.

Both string and hexadecimal pattern matching are included in an example YARA rule for malware detection. These rules are loaded at runtime to search folders like C:/Users for possible malware signatures. They are kept in a file called malware_rules.yar. These rules can be tailored by users to fit the particular dangers that are pertinent to their systems.

The system has certain restrictions even if it provides a number of helpful features. Since detection methods mostly rely on matching keywords and signatures, they may be less successful against threats that are obscured or unknown. Permission limitations may prevent some protected or system-level files from being accessed, which could compromise the thoroughness of scans. The present approach of finding firewall vulnerabilities depends on locating general rules, like those with "Allow" and "Any" conditions, which might not be able to detect all kinds of vulnerabilities. Additionally, the email alert system needs legitimate login credentials and may be impacted by email service provider security limitations.

This project has a lot of room to grow in the future. In the future, improvements will incorporate Scapy's real-time network packet inspection, machine learning algorithms for behavioural threat identification, user authentication, and encryption for logs and warnings. To improve usability and accessibility for a wider range of users, the system can also be further improved to enable cross-platform environments and be made available on mobile devices.

ACKNOWLEDGEMENT

My sincere gratitude goes out to everyone who helped and advised me throughout the creation of this project, "System Security Analyser Using Streamlit."

First and foremost, I would want to express my gratitude to, my faculty mentor and guide, for their insightful advice, helpful criticism, and unwavering support during the entire project. This work was greatly influenced by their knowledge and direction.

I am particularly appreciative of Sri Shakthi Institute of Engineering and Technology's Department of Computer Science and Engineering for providing the tools and facilities required to conduct this study. We would also want to thank the personnel and lab assistants who made sure the setting was suitable for testing and deployment.

I would like to thank my students and peers for their unwavering support, technical expertise, and recommendations, all of which were very beneficial throughout the design and development process.

Last but not least, our families has been the foundation of my path, and I am incredibly grateful for their consistent support, patience, and faith in my skills.

REFERENCES

[1] Sugandha Sharma [2018], ”Fighting Virus and Malware with Artificial Intelligence” available at https://www.insightssuccess.com/fighting-virus-and-malwarewith-artificial-intelligence/ [Accessed on 24 July 2018].

[2] Jason Brownlee [2016], “What is Deep Learning?” Available at: https://machinelearningmastery.com/what-is-deep-learning/ Accessed on 24 July 2018.

[3] P. Bisht V. Venkatakrishnan "Xss-guard: precise dynamic prevention of cross-site scripting attacks" in Detection of Intrusions and Malware and Vulnerability Assessment Springer pp. 23-43 2008.

[4] P. Laskov N. Šrndić "Static detection of malicious javascriptbearing pdf documents" Proceedings of the 27th Annual Computer Security Applications Conference. pp. 373-382 2011.

[5] KeterynaChumachenko [2017], ”Machine Learning Methods for Malware Detection and Classification” Processing of Kaakkois-Suomenammattikorkeakoulu, University of Applied Science in 2017.

[6] Jinpei Yan, Yong Qi and Qifan Rao [2018],”Detecting malware with an ensemble method based on deep neural network” Proceeding on Security and Communication Networks Volume 2018, Article ID 7247095, https://doi.org/10.1155/2018/7247095.

[7] Karishma Pandey, Madhura Naik, Junaid Qamar ,Mahendra Patil (2015), Spyware Detection using Data Mining, International Journal of Engineering and Techniques.

[8] Ms. Milan Jain, Ms. Punam Bajaj (2014), Malicious Code Detection through Data Mining Techniques, International Journal of Computer Science & Engineering Technology (IJCSET).

[9] Saba Arshad, Abid Khan, Munam Ali Shah, Mansoor Ahmed (2016), Android Malware Detection & Protection: A Survey, (IJACSA) International Journal of Advanced Computer Science and Applications.

[10] Z. Bakdash, Steve Hutchinson, Erin G. Zaroukian, Laura R. Marusich, Saravanan Thirumuruganathan , Charmaine Sample, Blaine Hoffman , and Gautam Das, Malware in the future? forecasting of analyst detection of cyber events Jonathan, University of Texas Dallas Dallas, TX, USA.

[11] Niklas Lavesson, Martin Boldt, Paul Davidsson, Andreas Jacobsson (2009), Learning to detect spyware using end user license agreements, Springer.

[12] Androutsopoulos I, Paliouras G, Karkaletsis V, Sakkis G, Spyropoulos CD, Stamatopoulos P (2000), Learning to filter spam Email: a comparison of a naive bayesian and a memory based approach.

[13] Kirti Mathur (2013), A Survey on Techniques in Detection and Analyzing Malware Executables, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4.

[14] M. G. Schultz, E. Eskin, E. Zadok and S. J. Stolfo (2001), Data Mining Methods for Detection of New Malicious Executables, Proceedings of the 2001 IEEE Symposium on Security and Privacy, IEEE Computer Society.

[15] Parisa Bahraminikoo (2012),Utilization Data Mining to Detect Spyware, IOSR Journal of Computer Engineering (IOSRJCE), Volume 4, Issue 3.