International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 38, No. 1, February, 2024, pp. 269-275
Journal homepage: http://iieta.org/journals/ria

An Innovative Keylogger Detection System Using Machine Learning Algorithms and Dendritic Cell Algorithm

Soham P. Chinchalkar, Rachna K. Somkunwar*

Department of Computer Engineering, Dr. D.Y.Patil Institute of Technology, Pimpri, Pune 18, India

Corresponding Author Email: rachnasomkunwar12@gmail.com

https://doi.org/10.18280/ria.380128

Received: 4 October 2023
Revised: 1 November 2023
Accepted: 29 November 2023
Available online: 29 February 2024

Keywords:
Dendritic Cell Algorithm (DCA), Naïve Bayes, Support Vector Machine (SVM), keylogger, typing speed

ABSTRACT

Every computer user deals with serious privacy and security challenges. Keyloggers are a type of software malware that records keystroke events from the console and saves them to a log file. It allows to obtain sensitive information like passwords, PINs, and usernames and communicates with vengeful attackers without attracting the attention of users. Keyloggers are also types of session hijackers that record user keystrokes made on the computer to steal any sensitive information from the system. Keyloggers are the most dangerous and covert malware for our system since they are difficult to detect because they run in the background of the computer. The primary issue with keylogger detection in a system is its time-consuming nature and its reliance on a particular type of input traffic behaviour. Keyloggers can be prevented using antiviruses, but, cannot be detected once they entered into the system. We proposed a system that combines Dendritic Cell Algorithms (DCA) and Machine Learning Algorithms (MLA) to address these problems. Our system can accurately detect a software keylogger if it is present which is based on the rate at which inputs are given to the system. The best accuracy was attained by our hybrid SVM-NB-DCA and SVM-DCA approach, with accuracies of 99.8% and 96%, respectively. Hence, results have shown that our hybrid system is effective and accurate for keylogger detection.

1. INTRODUCTION

Today's information technology industry is evolving swiftly. Cyber professionals find it extremely difficult to keep their privacy and security. Research indicates that there's a rapid increase in the quantity of new malware infections. Any illicit conduct that primarily involves the use of a computer for commission and theft is referred to as cybercrime. According to the U.S. Department of Justice, cybercrime now encompasses any unlawful behaviour where evidence is stored on a computer. Cybercrimes, which are growing more common, include computer-commissioned crimes such as network intrusions and computer-based variants of already existing crimes such as theft of identity, harassment, recording keystrokes, and terrorism, which have all become severe challenges for both individuals and countries. Cybercrime will undoubtedly increase as technology becomes more embedded into people's daily lives.

A keylogger is either a virus or hardware that tracks keystrokes. captures monitors and records our keystrokes as we type. It uses a command-and-control (C&C) server to deliver data to a hacker. The hacker utilizes the usernames and passwords he or she finds by analysing the keystrokes to access secure systems. Keylogging refers to the potential of such vital information being leaked while we enter the information into the system. Software keyloggers are spyware that captures the user's keystrokes and sends them to the keylogger [1].

Keyloggers have been used for spying purposes ever since people started registering. Keyloggers were used between the mid-1970s and the mid-1980s for a variety of functions, including top-secret government work. The US Embassy and Consulate offices in Moscow and St. Petersburg used IBM Electric typewriters, which were identified by Soviet undercover agents in the 1970s. The development and use of commercial keyloggers peaked in the latter part of the 1990s when numerous new devices entered the market at once, though there have been several different kinds of keylogging. Since then, there have been hundreds of corporate keyloggers available for purchase, each targeted towards a different audience and available in a variety of languages [2].

Keyloggers have become known as one of the major threats to security and privacy in recent years. A keylogger is a concealed spying program that captures user activities on the computer in a variety of ways including voice, screen, keyboard, and mouse logging, all while remaining fully undetectable. Although there are many beneficial uses for key-loggers, because of the exponential growth of Internet usage, their negative uses significantly surpass their positive ones. Key-loggers are now a significant threat to a computer's privacy and security because of their increased effectiveness [3].

Anti-virus software has become more difficult to identify the system's keyloggers. Many immune concepts have been extracted and used to develop Artificial Immune Systems (AISs) for different types of applications such as clustering, classification, and pattern recognition [4]. Evolutionary Computing (EC) refers to algorithms that are based on biological evolution principles. The most exciting area of EC is the artificial immune system [5].

The different techniques are designed to find the keylogger on the system. These strategies assist in taking the immediate actions required to protect the system. The DCA identifies the keylogging process by examining correlations between keylogging, file access, and network communication parameters. The study is being conducted with the assistance of DCA, which has a higher detection rate and a lower false alarm rate. The necessity for this application was critical because keylogging was exploding between 2003 and 2006 [1].

Since 1990, researchers have been working on artificial immune systems, with the goal of using biological immune systems to encourage solutions to non-biological challenges. DCA is an immune-based algorithm used in the Artificial Immune System (AIS). It is based on how dendritic cells act, also known as the human body's intrinsic intrusion detection agents [2].

Keylogger detection is challenging because it operates in hidden mode. Software keyloggers can be found using several methods including anti-Hook approaches, dendritic cell technique, bot identification, safe access to password-protected accounts, and spyware detection.

The bulk of keylogging detection approaches use signature and behaviour-based detection. The main disadvantage of the signature-based approach is that it cannot detect young Keyloggers. A technique known as behaviour-based detection examines the incoming file signature to determine the behaviour of the corresponding software. Some drawbacks of the keylogger detection system are shown in Figure 1.

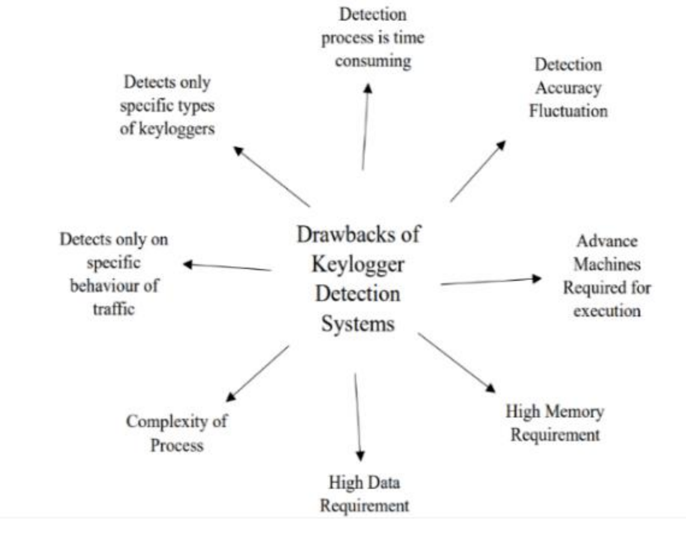
Detection process is time consuming
Detection Accuracy Fluctuation
Detects only specific types of keyloggers
Drawbacks of Keylogger Detection Systems
Detects only on specific behaviour of traffic
Advance Machines Required for execution
Complexity of Process
High Memory Requirement
High Data Requirement

Figure 1. Drawbacks of keylogger detection systems

The prevention of keyloggers accessing the system is 100% guaranteed by current antivirus programs like Norton 360 and Avast-one. Keyloggers can still access the system and carry out their evil deeds, nevertheless. According to statistics, Data breaches climbed by 2.6% in 2022, from $4.24 million in 2021 to $4.35 million. Therefore, a system that can identify keyloggers when they are present in the system is required. Checking Windows updates for security features and looking for strange apps running in the task manager are a couple of the straightforward methods now in use to assist in detecting whether a keylogger is there or not.

Our suggested method can identify the keylogger if it is present based on the rate at which inputs are entering the system. People can quickly become aware of the issue, stop keylogging, and save crucial information by employing this approach. Keyloggers are known to communicate their commands periodically. But what if a keylogger of this type is made to enter commands non-periodically? Such keyloggers can be found using our suggested technique. The findings of the proposed system demonstrate that it is an effective strategy for keylogger detection.

2. LITERATURE SURVEY

Ruhani and Zolkipli [6] have suggested an overview of keyloggers, their types, and their history. They also discussed keyloggers, both software, and hardware, and how they steal data from the machine. They have provided an outline of the safeguards and methods for system security. However, this paper's flaw is that no foolproof way to find system keyloggers has been provided. This study analyzes the future trends of keyloggers and serves as a guide for using them.

In the study [7], a combinatorial fuzzy inference system for keylogger detection was created, combining anomaly-based and signature-based techniques. This method's drawback is that it only detects replicated patterns; if the algorithm is unable to discover a certain pattern, keylogger detection will be hindered. Hence, our system is better as it is generalized, and not dependent on any specific situation.

By utilizing set theory and mathematical functions, Gu et al. [8] of this work have developed the deterministic form of an algorithm, referred to as Deterministic Dendritic Cell Algorithm (DCA). Additionally, they carried out runtime analysis and offered a clear explanation of the technique. Additionally, this approach gives DCA the tools to undertake continuous and periodic analysis. As a result, the algorithm is effective for detection tasks. This algorithm's drawback is that the system should be able to react to changes that occur in real-time while performing the detection.

Elisa et al. [9] have developed a method for mapping values using the K-Means clustering algorithm and the DCA. This study used the KDD99 dataset, and the findings showed increased classification accuracy. This approach's flaw is that real-time data sets were not used to test it, especially in terms of how well it performed when traffic behaviour changed quickly over time.

The main issue for an intrusion detection system, according to the authors' theory [10], is to identify and suppress unusual traffic patterns before they hurt the system. The DCA's principal operation was to determine the output cumulative values from the input signals using the weighted sum function. Users or empirical data are utilized to generate the weights for this function. The KDD99 dataset was used to validate and assess the methodology, and effective results were produced. In terms of true positives, false positives, and total classification accuracy, the outcome of the aforementioned strategy is quite effective. The initial work needs to be created in real-time from the data stream, and larger data sets need to be used for additional validation.

Some suggestions for service denial were given [11]. A

DOS is a type of assault that can be employed on systems. The system divides traffic into two categories: normal traffic and DoS attack traffic. To detect these attacks, the Dendritic Cell Algorithm, based on the Artificial Immune System, was used. The NSL-KDD dataset was utilized to analyse the detection system, and the results showed that it was extremely effective and successful in accurately detecting attacks. The negatives include the need for real-time DOS attack detection as well as the need to examine the effects of changing K-value and DCA lifespan on performance.

Navarro et al. [12] discussed kernel-level keyloggers. Keyloggers that operate at the kernel level represent an imminent threat to the security of current systems. Keyloggers installed at the kernel level have full access to the kernel code, data, and resources. To more effectively close the semantic gap between the operating system and architectural layers, a collaborative approach was developed. Operating system kernel integrity is safeguarded. This cooperative strategy may be a research direction for more effective OS kernel integrity protection.

Aslam et al. [13] suggested the anti-hook strategy while taking into account the operation of keyloggers for security and protection. The hijacking process will be able to be found and stopped by an anti-keylogger. The hidden files are also affected by this operation. Keylogging detection at the client level is carried out utilizing a hook-based method. When any unusual behaviour is detected in the input data, it checks all the hooks and sends out a threat.

Existing Systems

Sreenivas and Anitha [14], Keylogger detection was based on traffic generated into the system i.e., Traffic Analyses for Keylogger Detection, or TAKD. The keylogger starts communication with the server and enters commands so that user keystrokes can be recorded. This system examined the commands and discovered that the keylogger commands were periodic. As a result, it stopped and recognized keylogging traffic whenever four or more consecutive commands had the same gap in time between them. However, the irregular traffic patterns make it impossible for this system to function.

Using naive Bayes to detect spam emails the naive Bayes classifier was used in Rusland et al. [15] to categorize the emails as spam and ham. The results of the system have great accuracy for classifying emails as spam and ham, and how effectively this classification algorithm categorizes the dataset.

A dynamic Taint Analysis system was found [16]. Kernel-level keyloggers are those that operate at the kernel level of the operating system. These keyloggers modify keyboard data flow, which makes them challenging to detect. But by contaminating and observing the keystroke data, this system can find them. The experiment conducted using this framework can efficiently and accurately detect kernel-level keylogging activity. The significant restriction was that this proposed approach only applied to keyloggers at the kernel level.

The File Transfer Protocol (FTP) based detection method, also known as an anomaly-based detection methodology, is effective for hosts but only for keyloggers whose log transmission mechanism employs FTP [17].

TAKD [14] was one of the important works that demonstrated that keyloggers send commands periodically. When DCA [1] was used to detect keyloggers, the creation of antigens helped in analyzing traffic. These are the most significant points that we employed in our research.

3. BASIC CONCEPTS

Dendritic Cell Algorithm (DCA): It is a population dynamics method that employs Dendritic Cells to represent each agent shown in Figure 2. These cells undergo systemic alterations that cause the generation of antigens. The DCA is an algorithm based on population dynamics that is built on a theoretical framework of DC biology. Each agent is represented as a DC. Every cell has the ability to collect input signals (PAMP, danger, safe) that reflect changes in the monitored system's conditions and the antigens driving those changes. DCs generate cumulative output signals (CSM, semi-mature, and mature) by combining input signals. DC takes its time gathering antigens and messages. The CSM output signal (O1) rises in tandem with the input signal level experienced. The cell stops collecting signals and antigens until CSM reaches a "migration" threshold, at which point it is eliminated from the population in preparation for antigen presentation. For each antigen to present in a variety of settings, it is sampled many times.

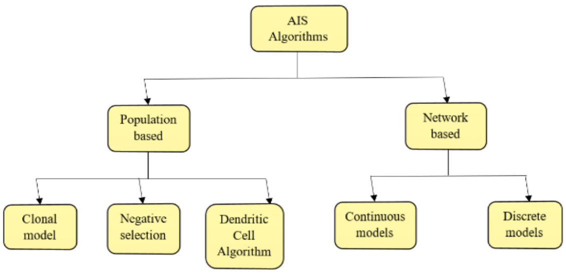

Figure 2. Classification of AIS

Artificial Immune System (ARTIS): ARTIS is made up of many algorithms that share many characteristics with the human immune system [18], as well as being widely distributed and remarkably adaptable [19]. Biological phenomena serve as the foundation for these systems, which aids in issue-solving. Additionally, it aids in safeguarding computers from any unauthorized invasions like keyloggers [20].

Antigens: When commands are encountered in the system, antigens are generated, resulting in the creation of signals. The system alterations have been identified as antigens. The Antigens have seen alterations in the cell brought on by outside factors in biological processes. Antigens are the DCA's source of input, and the pre-processing of data and signals determines the output [21].

Anti-Keylogging: Anti-keylogging is the process of discovering and eliminating vulnerabilities and infected places in the system. Such tools also can disable any hidden keystroke logging software from a computer, if not completely erase it. Effectively controlling malicious users involves an anti-keylogger [22].

Support Vector Machine (SVM): An algorithm is utilized for both classification and regression. This algorithm's objective is to create the best line possible and categorize the space so that input data points can be added to classes. The primary goal of the technique known as SVM is to discover the best hyperplane in the space of N dimensions for partitioning data points into different feature space classes. The hyperplane aims to maintain a wide range of separation between the nearest points across different classes. The number of features influences the hyperplane's dimensions. If only two input characteristics are present, the hyperplane is simply a line. When three input features are present, the hyperplane becomes a two-dimensional plane [23].

Naïve Bayes: This machine learning algorithm gives the prediction based on the probability of an object. Multinomial Naive Bayes, Bernoulli Naive Bayes, and Gaussian Naive Bayes are types of Naïve Bayes classifiers. This algorithm is based on the Bayes theorem with strong independent assumptions. Naive Bayes classifiers are simple probabilistic classification methods that rely on the theory of Bayes. Naive Bayes assumes that the attributes used to predict a value are independent of each other, which is frequently valid in real-world learning challenges. Regardless of this assumption, the naive Bayes classifier is extensively utilized for its efficiency and performance in a number of real-world circumstances [24].

4. PROPOSED SYSTEM

Keyloggers are highly advanced malware that record every keystroke performed on the computer, enabling attackers to stealthily and without the message owner's knowledge gather vast amounts of extremely sensitive data. To prevent data loss and the leakage of sensitive information, it's critical to recognize keyloggers. Antivirus software can employ heuristics and behaviour analysis to detect keyloggers; but, if the keylogger is not a known danger, antivirus or anti-malware software will fail to recognize it as a virus.

Dataset: Salthouse [25] investigates the relationship between typing speed and accuracy among typists with net rates ranging from 17 to 104 words per minute (wpm). We constructed a dataset of typing speeds and selected an average typing speed of 60 as our first experimental parameter. According to the investigations, performance ranged between 60 and 75 WPM, with typical IKIs of 140 ms [26, 27].

Confusion Matrix: To determine the level of accuracy of document classification systems, use the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) statistical measures. These elements combine to generate the Confusion Matrix. A confusion matrix is a table generated for a classifier on a binary dataset and used to describe how well it performs [28].

This matrix is based on the concept of True Positives (TP): Where both the actual and predicted results are positive. True Negatives (TN): When the prediction is "no" and the result is "yes". False Positives (FP): when the prediction is true but the result is false. False Negatives (FN): when the prediction is "no" and the result is "no".

Figure 3 depicts the proposed system’s architecture. In this work, we utilized a combination of methods to detect keyloggers.

In our method, both the user and the keylogger can enter data into our system, with the user providing input for regular operation and the keylogger providing input for duplicating the user's input's keystrokes.

Security measures for securing data entered into the system:

I. When initializing the system, use strong passwords.

II. Only higher-level officials should have access to sensitive system information.

III. Update devices to the most recent versions for optimal performance.

IV. Check for compatibility to avoid data loss and detection failures.

The proposed System executes the operations in the following steps:

(1) Firstly, Input is given to SVM. SVM categorizes input as greater than or equal to 60 and less than 60 based on the attribute of speed present in the dataset. Speed of input is referred to be normal user input if it is greater than or equal to 60, and suspicious input if it is less than 60. Keyloggers may be present in alleged input.

(2) Secondly, the dataset retrieved as suspected is given as input to the Naive Bayes algorithm. The naive Bayes classifier checks the time interval between each and every command that is getting inserted. Later, it classifies them as Periodic commands and Randomized commands and provides a probability for the presence of a keylogger accordingly.

(a) Periodic commands are commands that have the same time interval between them and if 4 or more consecutive commands have the same time interval between each other, then it certifies that there is a keylogger on the system. as the keylogger always inserts his commands periodically. These commands determine the high probability of the presence of a keylogger.

(b) Randomized commands are the commands that have no same time interval between them. These commands show a low probability of the presence of a keylogger.

(3) Now, Randomized commands and normal user commands will be given to the DCA algorithm. DCA is capable of generating high antigens and signals for short sentences which increases the keylogger detection rate if the keylogger is releasing un-periodic commands.

When we provide this combined input to DCA then DCA generates high antigens & signals and due to this keylogger is successfully detected.

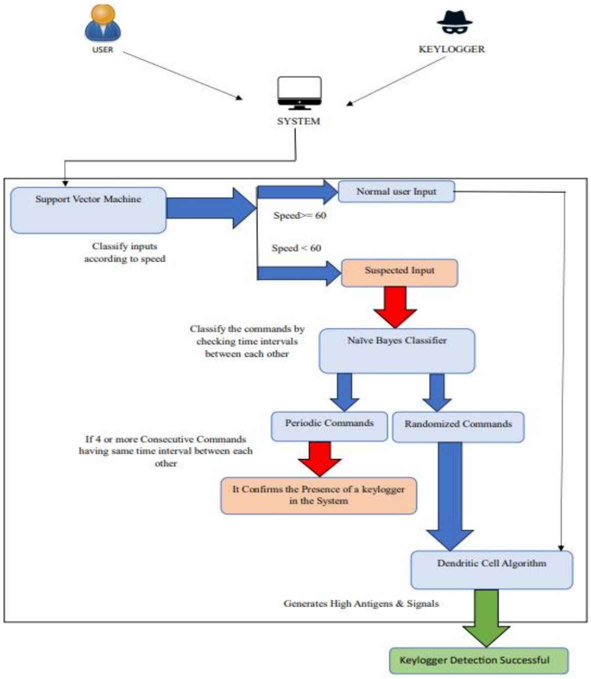
USER
KEYLOGGER
SYSTEM
Support Vector Machine
Classify inputs according to speed
Normal user Input
Speed>= 60
Speed < 60
Suspected Input
Classify the commands by checking time intervals between each other
Naïve Bayes Classifier
Periodic Commands
Randomized Commands
If 4 or more Consecutive Commands having same time interval between each other
It Confirms the Presence of a keylogger in the System
Dendritic Cell Algorithm
Generates High Antigens & Signals
Keylogger Detection Successful

Figure 3. Proposed architecture

5. RESULTS AND DISCUSSION

The performance and efficacy of our suggested system have been demonstrated in this section. The confusion matrix of SVM-DCA is shown in Table 1. Our assumption that the keylogger is present in the system and is accurate is expressed as a True positive value of SVM-DCA. We draw the conclusion that the higher the speed, the higher the detection since in our studies, the highest true positive result demonstrates that our detection accuracy is high for un-periodic traffic.

Table 1. Confusion matrix of SVM-DCA

SVM-DCA		
Predicted Values	Actual Values	
	Positive	Negative
Positive	1824	70
Negative	42	1395

Table 2 illustrates the confusion matrix for the SVM-NB-DCA. Our prediction that the keylogger is present in the system and is accurate is expressed as a True positive value of SVM-NB-DCA. This prediction comes true for the first time when we see the periodic keylogging commands that have previously confirmed the keylogger's presence in the system. Later, when the genuine positive value is high, it demonstrates that the system's detection accuracy is quite high, and we may draw the conclusion that, the slower the speed, the higher the detection. In the case of keylogger detection, a false negative value indicates that the keylogger is not there while, in fact, it is. Our output being 0 confirms that the detection is accurate.

Table 2. Confusion matrix of SVM-NB-DCA

SVM-NB-DCA		
Predicted Values	Actual Values	
	Positive	Negative
Positive	1324	2
Negative	0	583

A False Positive value of SVM-DCA and SVM-NB-DCA represents the prediction that the keylogger is present in the system while it is not. This value is the lowest in the tables of all the values that emphasize the system's authenticity. The confusion matrix's false positive and false negative values are critical. If these values increase, the reliability of the system can be called into doubt.

We have employed a number of evaluation measures, including True Positive Rate (X), False Positive Rate (Y), Precision (K), F-measure (F), Accuracy (A), and Matthew's Correlation Coefficient (Z), to provide additional validation. These are calculated using the confusion matrix's fields, as displayed in Tables 1 and 2.

X: The proportion of keyloggers in the system that have been accurately discovered is what is referred to as the True Positive Rate.

$$X=TP/TP+FN$$

Y: It is the percentage of keyloggers in the system that have been incorrectly identified.

$$Y=FP/FP+TN$$

K: It is a level of accuracy.

$$K=TP/TP+FP$$

F: It is described as the harmonic mean of recall and precision.

$$F=2\times K\times X/K+X=2\times TP/2\times TP+FN+FP$$

A (%): It is the proportion of keyloggers in the system that have been appropriately identified.

$$A\ (\%)=TP+TN/TP+FN+TN+FP\times 100$$

Z: It is used to determine how effectively machine learning algorithms perform in binary classification. It takes values from -1 to +1 and estimates the correlation between the actual and expected labels [28].

$$Z=\frac{TP\times TN-FP\times FN/}{\sqrt{}\ (TP+FP)\ (TP+FN)\ (TN+FP)\ (TN+FN)}$$

Table 3 provides a comparative analysis of algorithms based on parameters. The SVM-NB-DCA has the highest accuracy of 99.89 % with the lowest False Positive Rate and Highest value of Z i.e., 0.997.

Table 3. Comparison of proposed algorithms based on parameters

Algorithms	X	Y	K	F	Z	A (%)
SVM-DCA	0.977	0.047	0.963	0.969	0.931	96.63
SVM-NB-DCA	1	0.003	0.998	0.998	0.997	99.89

The model accuracy score is a statistic used to assess how effectively classifying models perform. Figure 4 depicts a graph of the accuracy rates for the SVM-DCA model for various kernels and default configurations. As a result, we derive the conclusion that our systems can accurately process and forecast values with the highest precision.

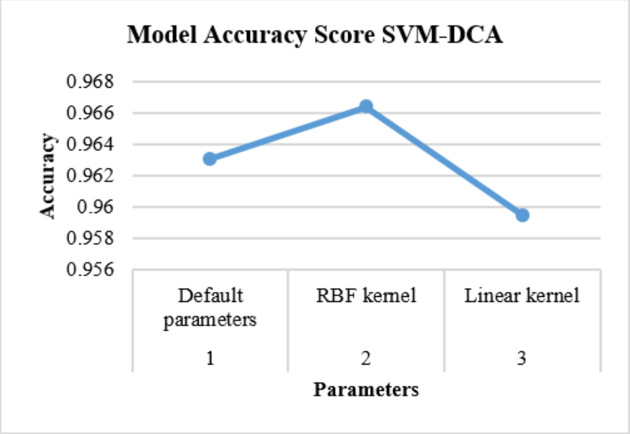

Figure 4. Model accuracy score of SVM-DCA

Regardless of how traffic behaves or what commands the keylogger transmits, our proposed method can detect the presence of keyloggers in the system. This ensures that the system can accurately identify keyloggers. The suggested hybrid SVM-NB-DCA and SVM-DCA approaches achieved the best accuracy, 99.8% and 96%, respectively, as shown in Figure 5. At this point, we derive the conclusion that our approach is more effective than current solutions like DCA, Traffic Analysis for Keylogger Detection (TAKD), and Keystroke Harvesting Malware (Klimax).

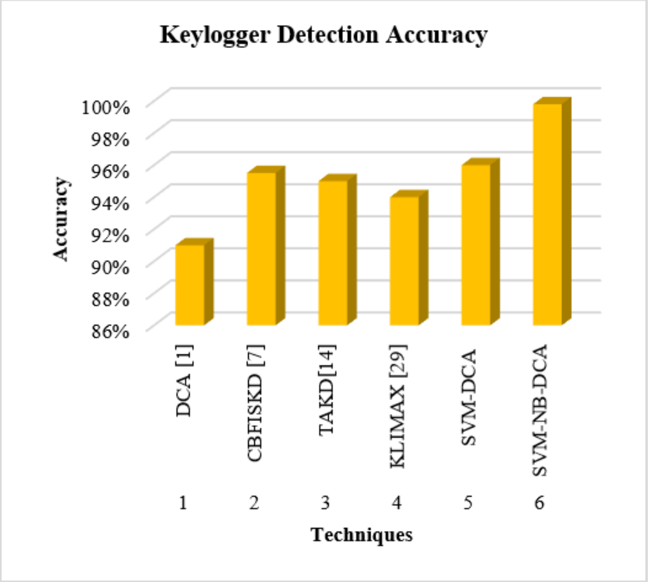

Figure 5. Keylogger detection accuracy, DCA [1] CBFISKD [7] TAKD [14] KLIMAX [29]

6. CONCLUSIONS

Keyloggers are particularly harmful since they have a high degree of accuracy. Many keyloggers operate silently for extended periods. Information security is all about preventing the mishandling, alteration, and destruction of information. In this work, we have used a hybrid approach which consists of Machine Learning algorithms and the Dendritic Cell Algorithm. This approach is effective and efficient in detecting the keyloggers present in the system. Our system has the benefit of being able to identify keyloggers in both periodic and non-periodic traffic patterns based on the speed of the input commands. Existing antivirus software and systems may implement this approach to detect keyloggers in the system more effectively. Our system's detection accuracy is 99.8%, and it also offers high accuracy for keylogger detection while enhancing process effectiveness and system security. The confusion matrix is provided in the final result, demonstrating the system's effectiveness. The proposed system's results show that it is an effective way of detecting keyloggers. Data breaches, which are among the most damaging aspects of the economy, can be managed or eliminated.

REFERENCES

[1] Fu, J., Liang, Y., Tan, C., Xiong, X. (2010). Detecting software keyloggers with Dendritic Cell Algorithm. In 2010 International Conference on Communications and Mobile Computing, pp. 111-115. https://doi.org/10.1109/CMC.2010.269

[2] Twycross, J., Aickelin, U. (2007). Biological inspiration for artificial immune systems. In International Conference on Artificial Immune Systems, pp. 300-311. https://doi.org/10.1007/978-3-540-73922-7_26

[3] Baig, M.M., Mahmood, W. (2007). A robust technique of anti key-logging using key-logging mechanism. In 2007 Inaugural IEEE-IES Digital EcoSystems and Technologies Conference, pp. 314-318. https://doi.org/10.1109/DEST.2007.371990

[4] Chelly, Z., Elouedi, Z. (2016). A survey of the Dendritic Cell Algorithm. Knowledge and Information Systems, 48: 505-535. https://doi.org/10.1007/s10115-015-0891-y

[5] Eiben, A.E., Smith, J.E. (2015). Introduction to evolutionary computing. Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/978-3-662-44874-8

[6] Ruhani, A.B.B., Zolkipli, M.F. (2023). Keylogger: The unsung hacking weapon. Borneo International Journal, 6(1): 33-43.

[7] Ayo, F.E., Awotunde, J.B., Olalekan, O.A., Imoize, A.L., Li, C.T., Lee, C.C. (2023). CBFISKD: A combinatorial-based fuzzy inference system for keylogger detection. Mathematics, 11(8): 1899. https://doi.org/10.3390/math11081899

[8] Gu, F., Greensmith, J., Aickelin, U. (2013). Theoretical formulation and analysis of the deterministic Dendritic Cell Algorithm. Biosystems, 111(2): 127-135. https://doi.org/10.1016/j.biosystems.2013.01.001

[9] Elisa, N., Yang, L., Qu, Y., Chao, F. (2018). A revised Dendritic Cell Algorithm using k-means clustering. In 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 1547-1554. https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00254

[10] Elisa, N., Yang, L., Naik, N. (2018). Dendritic Cell Algorithm with optimised parameters using genetic algorithm. In 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1-8. https://doi.org/10.1109/CEC.2018

[11] Dagdia, Z.C. (2019). A scalable and distributed Dendritic Cell Algorithm for big data classification. Swarm and Evolutionary Computation, 50: 100432. https://doi.org/10.1016/j.swevo.2018.08.009

[12] Navarro, J., Naudon, E., Oliveira, D. (2012). Bridging the semantic gap to mitigate kernel-level keyloggers. In 2012 IEEE Symposium on Security and Privacy Workshops, pp. 97-103. https://doi.org/10.1109/SPW.2012.22

[13] Aslam, M., Idrees, R.N., Baig, M.M., Arshad, M.A. (2004). Anti-hook shield against the software key loggers. In National Conference on Emerging Technologies, pp. 189-191.

[14] Sreenivas, R.S., Anitha, R. (2011). Detecting keyloggers based on traffic analysis with periodic behaviour. Network Security, 2011(7): 14-19. https://doi.org/10.1016/S1353-4858(11)70076-9

[15] Rusland, N.F., Wahid, N., Kasim, S., Hafit, H. (2017). Analysis of Naïve Bayes algorithm for email spam filtering across multiple datasets. In IOP Conference Series: Materials Science and Engineering, 226(1): 012091. https://doi.org/10.1088/1757-899X/226/1/012091

[16] Le, D., Yue, C., Smart, T., Wang, H. (2008). Detecting kernel level keyloggers through dynamic taint analysis. College of William & Mary, Department of Computer Science, Williamsburg, VA, Tech. Rep. WM-CS-2008-05.

[17] Sagiroglu, S., Canbek, G. (2009). Keyloggers: Increasing threats to computer security and privacy. IEEE Technology and Society Magazine, 28(3): 10-17. https://doi.org/10.1109/MTS.2009.934159

[18] Hofmeyr, S.A., Forrest, S. (2000). Architecture for an artificial immune system. Evolutionary Computation, 8(4): 443-473. https://doi.org/10.1162/106365600568257

[19] Timmis, J., Knight, T., de Castro, L.N., Hart, E. (2004). An overview of artificial immune systems. Computation in Cells and Tissues: Perspectives and Tools of Thought, 51-91. https://doi.org/10.1007/978-3-662-06369-9_4

[20] Hofmeyr, S.A., Forrest, S. (1999). Immunity by design: An artificial immune system. In Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2, pp. 1289-1296.

[21] Ali, K.B., Chelly, Z., Elouedi, Z. (2015). A new version of the dendritic cell immune algorithm based on the k-nearest neighbors. In Neural Information Processing: 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part I 22, pp. 688-695. https://doi.org/10.1007/978-3-319-26532-2_76

[22] Goring, S.P., Rabaiotti, J.R., Jones, A.J. (2007). Anti-keylogging measures for secure Internet login: An example of the law of unintended consequences. Computers & Security, 26(6): 421-426. https://doi.org/10.1016/j.cose.2007.05.003

[23] Sen, P.C., Hajra, M., Ghosh, M. (2020). Supervised classification algorithms in machine learning: A survey and review. In Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018, pp. 99-111. https://doi.org/10.1007/978-981-13-7403-6_11

[24] Kotsiantis, S.B., Zaharakis, I., Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging Artificial Intelligence Applications in Computer Engineering, 160(1): 3-24.

[25] Salthouse, T.A. (1984). Effects of age and skill in typing. Journal of Experimental Psychology: General, 113(3): 345-371. https://doi.org/10.1037/0096-3445.113.3.345

[26] Grudin, J.T. (1983). Error patterns in novice and skilled transcription typing. In Cognitive aspects of skilled typewriting, pp. 121-143. https://doi.org/10.1007/978-1-4612-5470-66

[27] Shaffer, L.H., Hardwick, J. (1968). Typing performance as a function of text. The Quarterly Journal of Experimental Psychology, 20(4): 360-369. https://doi.org/10.1080/14640746808400175

[28] Gandotra, E., Gupta, D. (2021). An efficient approach for phishing detection using machine learning. Multimedia Security: Algorithm Development, Analysis and Applications, 239-253. https://doi.org/10.1007/978-981-15-8711-5

[29] Ortolani, S., Giuffrida, C., Crispo, B. (2011). KLIMAX: Profiling memory write patterns to detect keystroke-harvesting malware. In Recent Advances in Intrusion Detection: 14th International Symposium, RAID 2011, Menlo Park, CA, USA, pp. 81-100. https://doi.org/10.1007/978-3-642-23644-0_5