DOI: 10.15849/IJASCA.240330.07

Int. J. Advance Soft Compu. Appl, Vol. 17, No. 1, March 2024
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Innovative Malware Detection: Practical Swarm Optimization and fuzzyKNN Model in Honeypot Environment

Heba Othman[1], Mwaffaq Abu Alhija[2], Mohammad A. Al Sharaiah[3]

[1,2] Department of Network and Information Security, Al-Ahliyya Amman University, Jordan
[3] Department of Data Science and Artificial Intelligence, Al-Ahliyya Amman University, Jordan
e-mail: m.sharaiah@ammanu.edu.jo

Abstract

Effective malware detection remains a critical challenge in cybersecurity. In this study, we propose an innovative method that combines swarm intelligence through Particle Swarm Optimization (PSO) with the fuzzy logic of the fuzzyKNN model, resulting in an adaptive and efficient malware detection system. Utilization of PSO assists in the selection of an optimal feature set from the malware dataset improving the performance of the fuzzyKNN model. create a secure and controlled environment for collecting diverse malware samples, we employ a honeypot. This controlled setting allows us to train our model without posing any risks to real operational systems. Conducted extensive tests to evaluate the effectiveness of our proposed methodology, comparing it against standard detection techniques. Our findings demonstrate the PSO-fuzzyKNN approach significantly enhances the accuracy of malware detection, outperforming traditional methods.contributes to advancement of malware detection technologies, offering a robust solution for addressing the evolving challenges posed by malicious software.

Keywords: *PSO-fuzzyKNN-based, fuzzyKNN model, cybersecurity, malware dataset, accuracy*

1. Introduction

In recent years, the increasing sophistication of malware has presented substantial challenges to cybersecurity. As cyber threats continually evolve, traditional approaches to malware detection struggle to keep up. To address the challenge, researchers and cybersecurity experts are exploring innovative methodologies to enhance detection capabilities and improve the resilience of security systems. The study focuses on integrating two advanced techniques,

Received 10 October 2023; Accepted February 2024

Practical Swarm Optimization (PSO) and the fuzzy k-Nearest Neighbors (fuzzyKNN) model, in the dynamic context of a Honeypot environment. Honeypots, intentionally vulnerable systems designed to attract and analyze malicious activities, offer an ideal environment for assessing the effectiveness of advanced malware detection mechanisms.

While traditional signature-based detection methods remain effective against known threats, they fall short when combating the rapid proliferation of polymorphic and zero-day malware. Consequently, there is a growing demand for adaptive and intelligent detection mechanisms capable of discerning subtle patterns indicative of malicious behavior. Swarm Optimization, inspired by collective intelligence in natural systems, and fuzzyKNN, incorporating fuzzy logic into the k-Nearest Neighbors algorithm, show promise in achieving the adaptability. The persistent advancement of malware threats has posed significant challenges to the cybersecurity community, prompting a shift towards machine learning techniques to develop more effective and adaptive malware detection solutions.[1]

The integration of Practical Swarm Optimization (PSO) and the fuzzy k-Nearest Neighbor (fuzzyKNN) model in a honeypot environment stands out among emerging approaches as a promising solution for malware detection. The innovative combination harnesses swarm intelligence and fuzzy logic to enhance detection accuracy and efficiency, as noted by [1].exemplified in Figure 1, affirm the enduring relevance and efficacy of Practical Swarm Optimization across various domains. These works highlight the adaptability of PSO in optimizing diverse and dynamic systems, underscoring its practical applicability in real-world scenarios. Furthermore, research by [2] underscores the ongoing evolution of PSO algorithms, showcasing their resilience and efficiency in addressing contemporary optimization challenges. These recent references underscore the enduring significance of PSO as a versatile and powerful optimization technique inspired by collaborative principles found in nature.

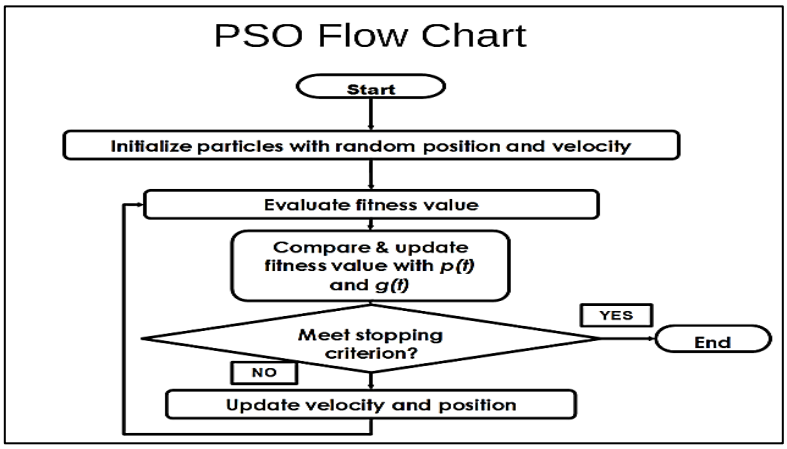

Figure 1: Particle Swarm Optimization [3]

The extensively used instance-based classification technique, is the k-Nearest Neighbor (KNN) algorithm (depicted in Figure 2), functions by determining the class of an unfamiliar data point based on the classes of its k-nearest neighbors within the feature space. Conversely, the Fuzzy k-Nearest Neighbor (fuzzyKNN) augments KNN by integrating fuzzy logic. The integration allows data points to be associated with multiple classes, exhibiting diverse levels of membership. The incorporation of fuzzy logic bestows a valuable classification ability,

proving particularly advantageous for addressing uncertainties and inaccuracies commonly encountered in real-world datasets [3].

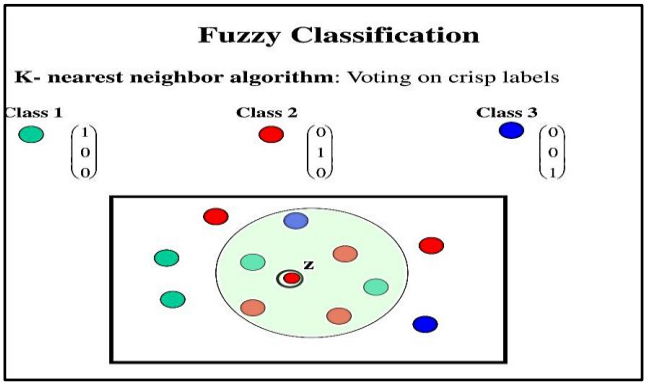

Figure 2: k-Nearest Neighbor (KNN) algorithm [3]

A honeypot, designed as a deceptive system to attract and ensnare malicious actors, serves as a crucial tool for security analysts to observe and analyze threat behaviors without compromising actual production systems. The integration of the PSO-fuzzyKNN model into a honeypot environment becomes instrumental in enhancing detection capabilities, enabling real-time analysis of suspicious activities, and early identification of new and emerging malware variants [4].Examining research on deep learning algorithms emphasizes the pivotal role of extensive datasets in achieving reliable outcomes. Deep learning architectures, primarily relying on supervised learning, necessitate a large number of labeled instances for effective model training [61]. The research underscores the challenge posed by small datasets, hindering the model's ability to grasp essential features adequately, leading to less reliable outcomes.

Current detection mechanisms face evident limitations in dynamic environments like Honeypots, intentionally exposed systems designed for analyzing malicious activity. Traditional approaches struggle to match the evolving tactics of cyber adversaries, resulting in a widening detection gap. Static signature-based methods lack adaptability to handle the dynamic nature of modern malware, often failing against polymorphic variants and novel attack vectors. The prevalence of false positives inundates security teams with irrelevant alerts, while sophisticated threats lead to false negatives. Additionally, conventional methods lack the nuanced context awareness required to differentiate between normal and malicious behaviors in intricate Honeypot environments. As cyber threats grow in volume and diversity, scalability challenges further strain existing detection systems' ability to process vast amounts of real-time data.

The exploration of Practical Swarm Optimization (PSO) and the fuzzy k-Nearest Neighbor (fuzzyKNN) model within a honeypot environment for advanced malware detection is motivated by the imperative need for more robust and adaptive solutions in the face of the continually evolving malware landscape. Traditional signature-based and heuristic detection methods exhibit limitations in identifying new and unknown malware variants, necessitating innovative approaches to effectively discern and mitigate emerging threats [19].

Practical Swarm Optimization (PSO), inspired by collective intelligence in social organisms, emerges as a promising solution for optimization in complex and ever-changing settings. Its fusion with the fuzzy k-Nearest Neighbor (fuzzyKNN) model equips the detection

system to effectively manage data uncertainty and imprecision, enhancing capabilities for nuanced and precise malware classification [19]. Recent references underscore the growing interest and research efforts directed at applying swarm intelligence and fuzzy logic-based approaches to overcome malware detection challenges. These studies highlight the efficacy of the proposed PSO-fuzzyKNN model in improving the accuracy and efficiency of malware detection systems, positioning it as a promising direction to address evolving cybersecurity threats.

The research on Innovative Malware Detection, utilizing Practical Swarm Optimization (PSO) and the fuzzy k-Nearest Neighbor (fuzzyKNN) Model within a Honeypot Environment, is driven by several key objectives. The first goal is to craft a Hybrid PSO-fuzzyKNN Model, leveraging the swarm intelligence of PSO to optimize the parameters of the fuzzyKNN algorithm, thereby enhancing the accuracy and efficiency of malware detection [20].

The second objective involves the implementation of the developed PSO-fuzzyKNN model in a Honeypot Environment. The integration aims to enable real-time analysis of suspicious activities, providing a robust mechanism for the detection of previously unknown and emerging malware variants. Additionally, the research endeavors to evaluate and compare the performance of the proposed malware detection system, utilizing metrics such as detection rate, false positive rate, precision, and recall. A comprehensive comparison with conventional signature-based and heuristic techniques will highlight the superior efficacy of the PSO-fuzzyKNN approach [21].

2. Literature Review

Malicious activities have been detected, primarily through the use of malicious spam campaigns [4]. These campaigns commonly exploit Microsoft Office files, enticing users to download and open corrupted files. Once opened, the malware manipulates users into enabling macros or exploiting vulnerabilities. The identified malware, Hancitor, is then either fetched from a command and control (C2) server or delivered from within an Office file. Upon activation, Hancitor proceeds to download its primary payload, often a Trojan such as Pony, Vawtrak, or DELoader. Hancitor utilizes various techniques, including DOC attachments exploiting Microsoft's dynamic data exchange (DDE) mechanism [5]. The method requires users to download the file and deliberately activate macros, often bypassing multiple security alerts. To facilitate this, malware authors create convincing lures to entice users into performing these actions.

Hackers are actively producing approximately 230,000 new malware samples each day, a number expected to rise in the future. Ransomware has swiftly emerged as a significant threat, with a staggering 4,000 ransomware attacks documented. The impact of ransomware spans from individual users to small businesses and large enterprises, potentially resulting in the loss of sensitive data, either temporarily or permanently [7]. Critical infrastructure is notably vulnerable, drawing the attention of those familiar with the extensive damage ransomware can inflict. The category of malware utilizes encryption modules to lock data, rendering it unusable for the victim, as emphasized by [8].

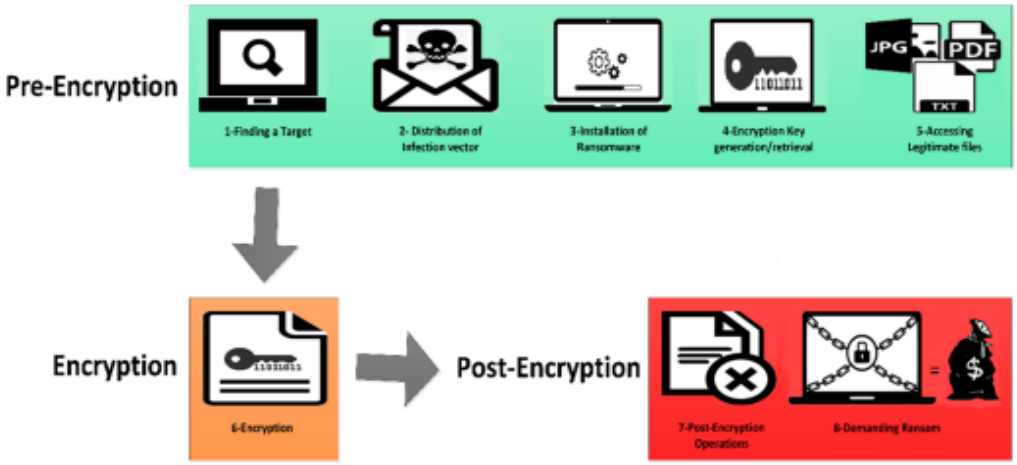

Figure 3: Ransomware attack phases [8]

The literature reveals the widespread impact of ransomware, affecting not only small businesses but major corporations such as FedEx, Nissan, and NHS organizations in the UK [9]. Kaspersky's report emphasizes the enduring prevalence of spam emails in phishing efforts. Symantec's Internet Security Threat Report for 2019 [10] highlights the vulnerability of supply chains, experiencing a 78% surge in attacks in 2019 compared to the previous year. The report also notes a fourfold increase in blocking 69 million cryptojacking incidents in 2018 compared to 2017. Small businesses face a substantial impact, with 40% falling victim to attacks in 2019, leading to the collapse of 60% due to economic losses. Accenture's findings indicate significant investments, with companies allocating $2.4 million for malware detection and defense against web-based attacks. Critical infrastructure has witnessed cyber turmoil, with recent instances involving state-sponsored attackers targeting industrial control systems. [6].

The contemporary threat landscape of malware is characterized by a dynamic spectrum of malicious software aiming to infiltrate computer systems and networks, causing harm, stealing sensitive information, or disrupting operations. Traditional signature-based antivirus solutions struggle to match the dynamic and polymorphic nature of malware, necessitating innovative detection techniques to stay ahead of cyber threats.

Advanced Persistent Threats (APTs) and Advanced Evasion Techniques (AETs) represent sophisticated malware variants designed for prolonged undetection. They utilize encryption, obfuscation, and anti-analysis techniques, posing substantial challenges to traditional security measures [23]. Ransomware remains a prominent and lucrative threat, encrypting critical data and demanding ransom payments. The use of cryptocurrency and ransomware-as-a-service (RaaS) models facilitates large-scale ransomware campaigns, emphasizing the need for robust cybersecurity measures [24].

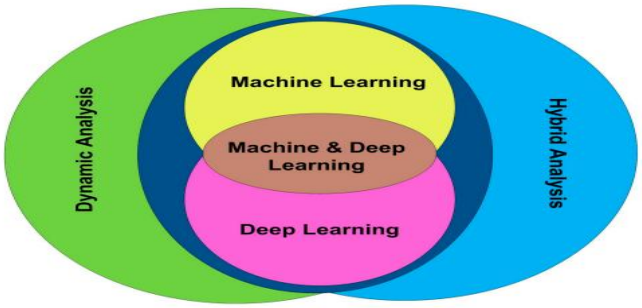

Figure 4: Ransomware analysis overview [26].

Hybrid analysis combines static and dynamic methods to achieve precise results. In a mobile-based ransomware detection instance, machine learning classifiers and statistical assessments were used before dynamic analysis. Fileless malware, residing in memory, poses challenges for traditional detection. IoT-based malware exploits vulnerable smart devices, creating botnets. Zero-day exploits target unknown vulnerabilities. The evolving malware landscape demands innovative approaches, such as Practical Swarm Optimization and fuzzy k-Nearest Neighbor models in a honeypot environment, to enhance detection and analysis efficiency [27-31].

Traditional cybersecurity methods, like signature-based and heuristic detection, encounter challenges against sophisticated threats, struggling with unknown or polymorphic malware and generating false positives [30][23][32]. Sandbox analysis aids in novel malware detection, but evasion by sophisticated malware is a concern [34]. Machine learning approaches, explored in a study [35], exhibit effectiveness against unseen threats but may suffer from false negatives or positives without sufficient training.

Swarm intelligence, observed in social organisms, results in complex group behavior. Examples include ACO, Bird Flocking, Bee Swarming, and Fish Schooling [39-40]. PSO, introduced in 1995, optimizes solutions by adjusting particle positions and velocities based on personal and global best experiences, effectively solving dynamic problems [41].

2.1 Principles of Practical Swarm Optimization (PSO)

Practical Swarm Optimization (PSO) is an optimization algorithm inspired by the collective intelligence observed in social organisms, such as birds flocking and fish schooling. Introduced by Kennedy and Eberhart in 1995, PSO has gained popularity as a robust and efficient optimization technique in various fields, including machine learning, engineering, and data mining. The principles of PSO revolve around the behavior and interactions of particles in a search space to find the optimal solution [42,43].

In PSO, a population of particles represents potential solutions to the optimization problem. Each particle is analogous to an individual in a swarm and is characterized by a position and velocity in the search space. The position of a particle corresponds to a potential solution, while the velocity determines the direction and magnitude of its movement in the search space.The fitness function evaluates the quality of each particle's position (solution) in the search space.

The objective is to minimize or maximize the fitness function, depending on the optimization problem's nature. The fitness function guides the particles to explore the search space efficiently and converges toward the optimal solution.

Each particle maintains its best position (solution) found during the search process. The position is known as the personal best (pBest). The pBest represents the individual's best performance in the optimization process.In addition to the personal best, PSO keeps track of the best position (solution) found by any particle in the entire swarm. The position is known as the global best (gBest). The gBest represents the best solution achieved by any particle in the swarm so far [44,45].

2.2 PSO for Feature Selection in Malware Detection

In malware detection, the optimization prowess of Practical Swarm Optimization (PSO) shines as it navigates the intricate task of feature selection. By framing feature selection as an optimization challenge, PSO dynamically explores varied feature combinations, iteratively adjusting particle positions to converge on an optimal subset. The process aims to maximize classification accuracy while minimizing computational complexity, [42]

The fuzzy k-Nearest Neighbor (fuzzyKNN) model, a sophisticated evolution of the traditional k-Nearest Neighbor (kNN) algorithm, is tailored for cutting-edge malware detection. FuzzyKNN's integration of fuzzy logic effectively addresses the complexities inherent in real-world malware scenarios, embracing uncertainties and imprecise boundaries. At its core, fuzzyKNN assigns a fuzzy membership function to each data point, capturing partial membership across multiple classes based on proximity to class prototypes. Leveraging distance metrics like Euclidean distance, fuzzyKNN calculates similarities during classification, employing a weighted voting mechanism to determine nuanced and precise class membership.

Crucial to its effectiveness is data preprocessing, including tasks like handling missing values, normalizing data, and selecting pertinent features. During the training phase, fuzzyKNN constructs prototype sets for each class based on labeled data, while the classification phase utilizes fuzzy membership degrees to assign final class labels. By leveraging fuzzy logic's strengths, fuzzyKNN adeptly handles uncertainties, noise, and overlapping boundaries—essential attributes for robust malware detection. Integration into a honeypot environment alongside Practical Swarm Optimization significantly amplifies the accuracy and effectiveness of malware detection and analysis.[43]

A honeypot functions as a distraction system or network, intentionally designed to mimic vulnerable targets and attract cybercriminals. By simulating alluring yet unprotected environments, honeypots serve as an early warning system, detecting and diverting attackers before they reach critical assets. Crucially, these deceptive setups gather threat intelligence, offering insights into attackers' tactics and aiding in security enhancement. Honeypots enable the study of real-world attack scenarios, informing defense strategies. Integrating Practical Swarm Optimization and the fuzzy k-Nearest Neighbor model in a honeypot environment elevates malware detection capabilities, providing organizations with a comprehensive solution to analyze, defend, and neutralize cyber threats proactively.

Honeypots, integral to innovative malware detection systems, manifest in diverse types and deployment strategies, each catering to specific objectives and offering distinct advantages.

High-Interaction Honeypots furnish realistic environments for in-depth attacker interaction, demanding more resources. Conversely, Low-Interaction Honeypots emulate specific services, capturing information with lower resource consumption. Production Honeypots fortify live networks, alerting administrators to potential threats, while Research Honeypots, in controlled environments, gather threat intelligence. Decoy Honeypots divert attackers from critical assets, providing a defensive layer, while High Interaction with Limited Reach Honeypots focus on specific network segments. The selection depends on the malware detection system's goals, enhancing the comprehensive approach of Practical Swarm Optimization and the fuzzy k-Nearest Neighbor model in a honeypot environment.[44]

The PSO-fuzzyKNN model architecture introduces an innovative paradigm for malware detection, amalgamating Practical Swarm Optimization (PSO) and the fuzzy k-Nearest Neighbor (fuzzyKNN) model within a honeypot environment. The architecture synergizes the unique strengths of both PSO and fuzzyKNN, optimizing feature selection, managing uncertainty, and elevating the accuracy and efficiency of malware detection. The comprehensive workflow and key components of the PSO-fuzzyKNN model architecture are outlined below:

Following feature selection, a fuzzy membership function is generated for each class based on the selected features. These functions map data points to their degrees of membership in multiple classes, providing nuanced insights into the truthfulness of data point classifications.Leveraging the PSO-optimized feature subset and corresponding membership functions, the fuzzyKNN model undergoes training. Fuzzy logic-based inference allows the model to classify new data points, accommodating partial memberships and handling uncertainty during classification.Strategically placed honeypots within a controlled environment attract and capture malware samples. The captured data continually updates and refines the PSO-fuzzyKNN model, enhancing its proficiency in detecting and analyzing emerging malware threats.

The training of the PSO-fuzzyKNN model using honeypot data is a pivotal stage in crafting an advanced malware detection system. The amalgamation of Practical Swarm Optimization (PSO) and the fuzzy k-Nearest Neighbor (fuzzyKNN) model within a honeypot environment significantly elevates the model's precision in distinguishing between malicious and benign activities. The training protocol encompasses strategic honeypot deployment, data preprocessing, PSO-driven feature selection, and fuzzy membership function generation.

Honeypots strategically placed in controlled environments lure and capture malware samples, mimicking vulnerable systems to record attacker interactions. Collected honeypot data undergoes preprocessing to rectify noise, missing values, and irrelevant information. PSO optimally selects features crucial for malware classification, aiming for accuracy while minimizing computational complexity.Post-feature selection, fuzzy membership functions are generated, mapping data points to degrees of membership across classes. The PSO-optimized features and membership functions then train the fuzzyKNN model. Fuzzy logic-based inference ensures nuanced classification, accommodating partial memberships and handling uncertainty.

3. Methodology

The model undergoes training and evaluation using preprocessed datasets for training and testing. Feature optimization employs Particle Swarm Optimization (PSO) to enhance the model's performance by iteratively improving features. PSO iterates to find the optimal feature combination, improving predictive accuracy. Subsequently, a fuzzyKNN model is trained on these optimized features, leveraging fuzzy logic for datasets with fuzzy class boundaries. Model evaluation utilizes metrics like ACC, PRE, REC, and F1-score, providing a comprehensive performance overview for identifying areas of improvement and guiding future research directions.

3.1 Data Collection

The N-BaIoT dataset, developed by [42], encompasses network traffic data from nine devices intentionally infected with the Mirai and BASHLITE botnets. The deliberate infection aimed to capture authentic malicious traffic data. Comprising 115 features, the dataset was gathered using port mirroring, yielding over 1.3 million samples categorized into Mirai attack, benign, and Gafgyt (BASHLITE) attack groups. The extensive dataset serves as a valuable asset for cybersecurity research, facilitating the creation of machine learning models capable of discerning various botnet attacks and benign traffic data, as illustrated in Figure 5.

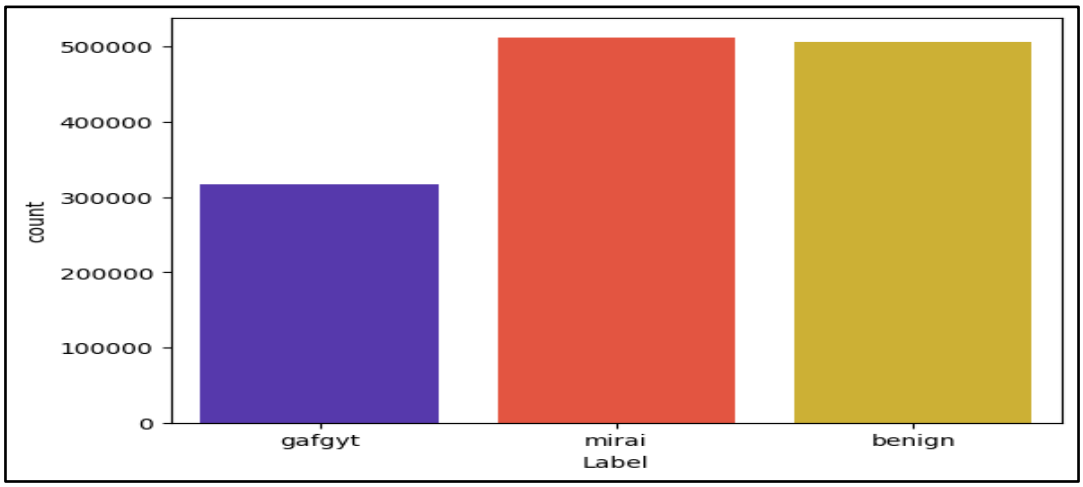

Figure 5**:** Distributing Data

Honeypots were employed to capture the N-BaIoT data. Depicted in Figure 6, the testbed for the Bot-IoT dataset consists of a cluster of both malicious and benign virtual machines

(VMs) connected to LAN and WAN interfaces. The setup serves as a valuable resource for the analysis and modeling of security risks.

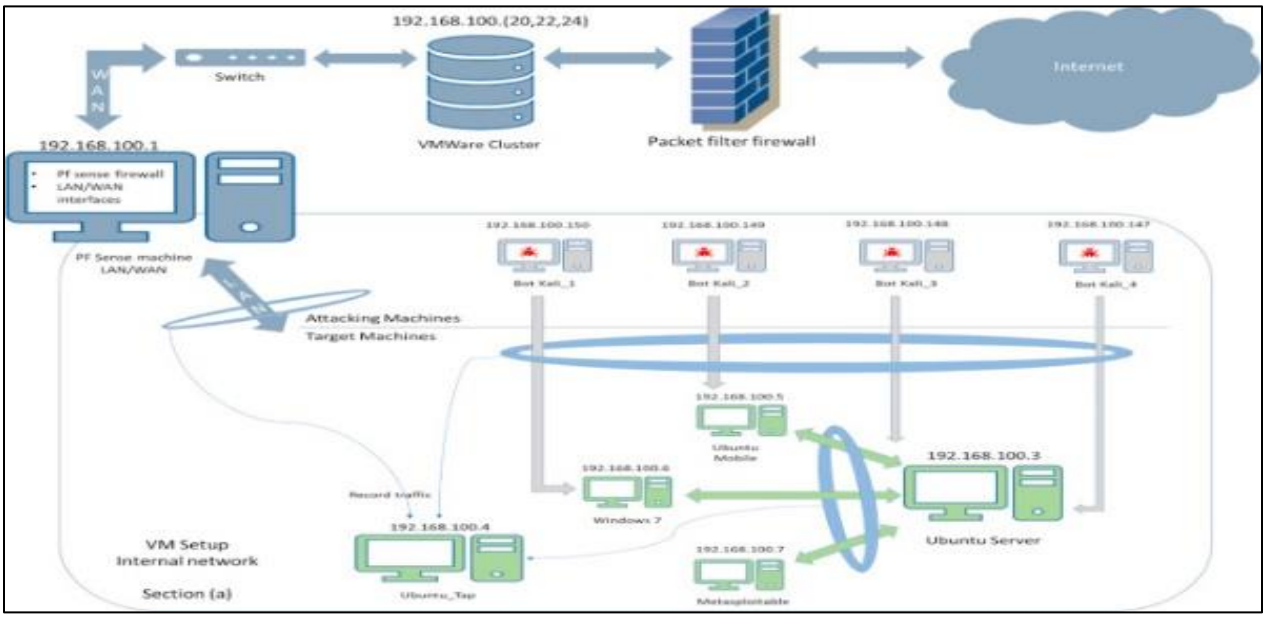

Figure 6: Bot-IoT honeypot.

3.2 Data Preparation

3.2.1 Handling Missing Values

The dataset exhibits completeness with no null or missing values, indicating a lack of superfluous data. The absence of missing values obviates the need for imputation or data filling. To further enhance the dataset, the elimination of duplicate data has been implemented. While the dataset is free from missing values, optimization processes, such as feature extraction, label encoding, and data normalization, are still required to prepare it for comprehensive analysis.

3.2.2 Label Encoding

Label encoding is employed to convert categorical data into numerical form by assigning unique numerical labels to categories. In the context, "benign," "gafgyt," and "mirai" are represented as numerical values (0, 1, and 2). The transformation facilitates the interpretation of categorical input as numerical data, simplifying the model fitting process. The frequencies of 1 and 0 are 828,783 and 506,384, respectively, indicating a dataset skew where malware traffic (labeled as 1) surpasses benign traffic (labeled as 0). The imbalance in the dataset may impact the performance of machine learning models, necessitating additional measures for optimal handling.

3.2.3 Data Balancing

To utilize down samplings like random undersampling, oversampling, or both to fix the unbalanced dataset. The algorithm randomly undersampled the majority class (label 0) by picking 506,384 samples from the minority class (label 1). Created a balanced dataset with equal labels. Balancing the dataset can increase ML model performance by minimizing bias towards the dominant class and allowing the model to learn from both classes. To choose balanced data forms to train and test the FuzzyKNN classifier (illustrated in Figure. 7).

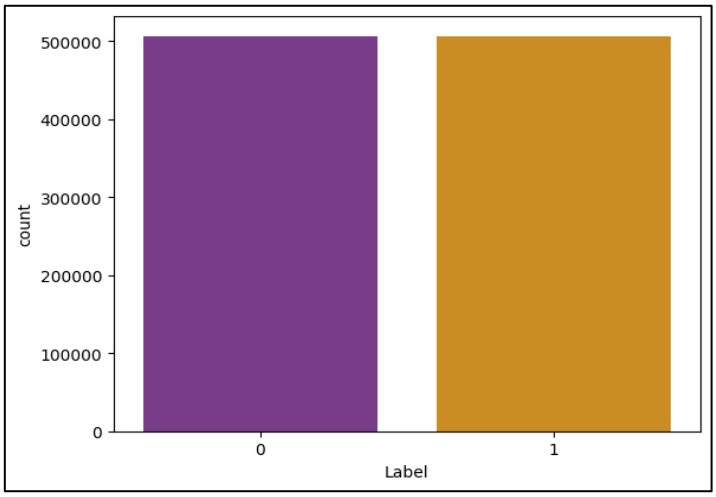

Figure 7**:** Balanced Data Distribution.

3.2.4 Data Normalization

Data normalization is an essential preprocessing step in machine learning and data analysis, aimed at transforming the data into a standardized format. One commonly used technique for normalization is called "Min Max Scaler." The method rescales the data to fit within a range of 0 to 1. Achieved by subtracting the minimum value of the feature and dividing it by the range of the feature..

3.2.5 Data Splitting

Training and testing datasets were split. The test size option is set to 0.2, meaning 20% of the dataset will be tested and 80% trained. To eliminate ordering bias, the data was randomly mixed before splitting. Splitting the dataset lets us train and test our ML model. In the work, PSO optimizes RF Classifier hyperparameters for binary classification. Optimization reduces validation dataset log-loss. An objective function, maximum of 10 iterations, population size of 20, and minimize parameter set to True initialize the PSO algorithm. PSO iteratively updates particle positions and velocities to obtain the best solution. Figure 8 shows the method converges to the ideal solution or reaches 10 iterations.

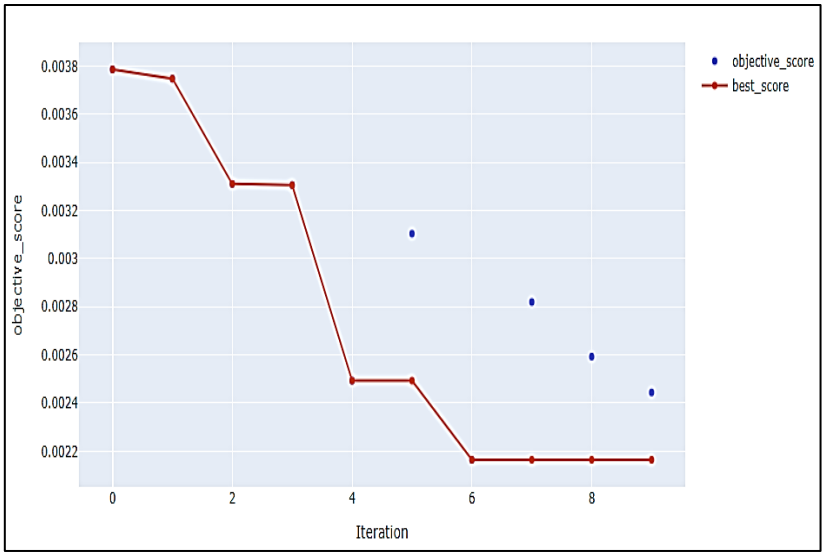

Figure 8: Optimization history plot.

To determine if 10 iterations are preferable, need additional detail and clarity regarding what is being optimized and how the objective value varies with iterations. Several iterations' "bitterness" relies on various factors:

1. Objective function: The optimization objective function matters. It convex? Many local minima? These questions can impact the number of iterations needed to find a good solution.
2. Halting criteria: The stopping criterion affects iterations. One could end if the goal function change is below a threshold or after a given number of iterations.
3. Cost-quality tradeoff: Iterations frequently improve solutions, but they take more time and memory.

Solution quality and computing cost usually clash. The log output shows that the objective value lowers and improves (smaller) with time, implying that additional iterations improve the solution. Improvements eventually slow down. Means that an acceptable stopping condition was reached, and more iterations would waste resources without enhancing the answer.

4. Experimental Evaluation and Performance Metrics

When evaluating the performance of an innovative malware detection system using Practical Swarm Optimization (PSO) and a fuzzyKNN (fuzzy k-nearest neighbors) model in a honeypot environment, several evaluation metrics can be employed to assess its effectiveness. Here are some commonly used metrics:

Detection Rate (DR) or True Positive Rate (TPR):

- Definition: The proportion of actual malware instances that are correctly identified by the system.

- Formula: $DR = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$ **……..…….(1)**

False Positive Rate (FPR):

- Definition: The proportion of non-malicious instances incorrectly identified as malware.
- Formula: $DR = \frac{\text{False Negatives}}{\text{False Negatives+True Positives}}$ **…….……(2)**

Precision:

- Definition: The accuracy of the system in correctly identifying malware instances among all instances labeled as malware.
- Formula: $\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$**……(3)**

Recall or Sensitivity:

- Definition: The ability of the system to identify all actual malware instances.
- Formula: $\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$**……..……(4)**

F1 Score:

- Definition: The harmonic mean of precision and recall, providing a balanced measure between the two.
- Formula : F1 Score = 2 X $\frac{\text{Precision x recall}}{\text{Precision+ recall}}$**…….(5)**

Accuracy:

- Definition: The overall correctness of the system in classifying both malware and non-malware instances.
- Formula: $\text{Accuracy} = \frac{\text{True Positives+True Negatives}}{\text{Total instances}}$**……..(6)**

The chosen metrics serve as a comprehensive yardstick for evaluating the malware detection system, encompassing crucial parameters like accuracy, precision, recall, and the delicate balance between false positives and false negatives. Metric selection hinges on aligning with the detection system's specific goals and priorities, considering the dataset's unique characteristics.The paper delineates the research methodology employed in identifying Internet of Things (IoT) malware, with a focus on the study's various approaches:

Dataset Collection Stage: The initial phase involves acquiring the N-BaIoT dataset, a compilation of real traffic data infected by Mirai and BASHLITE malware. The data capture utilizes a honeypot design, employing multiple virtual machines (VMs) connected to a network cluster.Data Preprocessing Stage, The raw data undergoes meticulous preprocessing to enhance its suitability for machine learning algorithms. Involves Removal or imputation of incomplete entries, Balancing the dataset to mitigate class bias, Normalization to standardize features,

optimizing machine learning techniques.Label encoding to convert categorical variables into numerical inputs, a prerequisite for machine learning algorithms.

5. Statistical & Performance Analysis

5.1 Sample Size Impact:

Throughout varying sample sizes of 60,000, 100,000, and 160,000 cases, the models consistently demonstrated robust performance. The accuracy (ACC) exhibited a range from 99.95% to 99.97%, showcasing the models' reliability as the sample size increased. Precision (PRE), Recall (REC), and F1-score consistently achieved 96% across most sample sizes and K values. Notably, the models exhibited high sensitivity and specificity in detecting both malware and normal occurrences. Importantly, as the sample size increased, the performance did not exhibit a significant decline; instead, certain performance indicators showed marginal improvements. These findings suggest that the FuzzyKNN model has the potential to scale effectively, maintaining its efficacy as the dataset expands.

5.2 Confusion Matrices Results

Confusion matrices show model performance for varied data forms and K values. The model predicts TP, TN, FP, and FN in each matrix. Figures depict K-value confusion matrices.

Dataset with 60000 samples (Figure 9.)

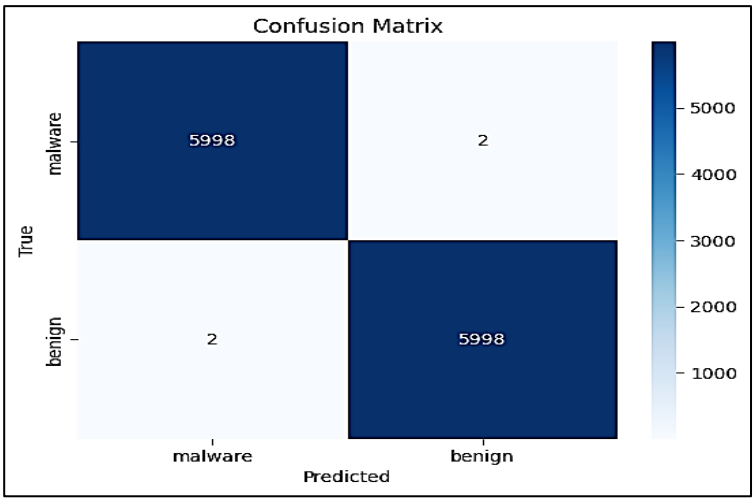

Figure 9: FuzzyKNN with 60000 samples and K=2.

5.3 Compared Times and Accuracy

Figure 9 illustrates a discernible trend regarding the interplay of training samples and the choice of K in Fuzzy K-Nearest Neighbors (Fuzzy_KNN), particularly in terms of

accuracy and training time. Notably, an expansion in the training sample set size from 60,000 to 100,000 correlates with enhanced accuracy and an increase in training time. Theoutcome aligns with expectations, as larger datasets inherently offer a more comprehensive representation of the problem space, bolstering prediction model accuracy. However, the trade-off involves heightened computational resource requirements and processing time.

Analyzing Figure 10 reveals insightful details regarding the impact of training samples and K in Fuzzy_KNN. With a dataset of 60,000 training samples, accuracy consistently ranged between 0.9995 and 0.9996, with training time varying across different K values. Conversely, with 100,000 training samples, a slight improvement in accuracy (0.9997 to 0.9998) was accompanied by a notable increase in training time (0.082 to 0.134). These specifics illuminate the intricate relationship between training sample size, accuracy, and training time in the context of Fuzzy_KNN.

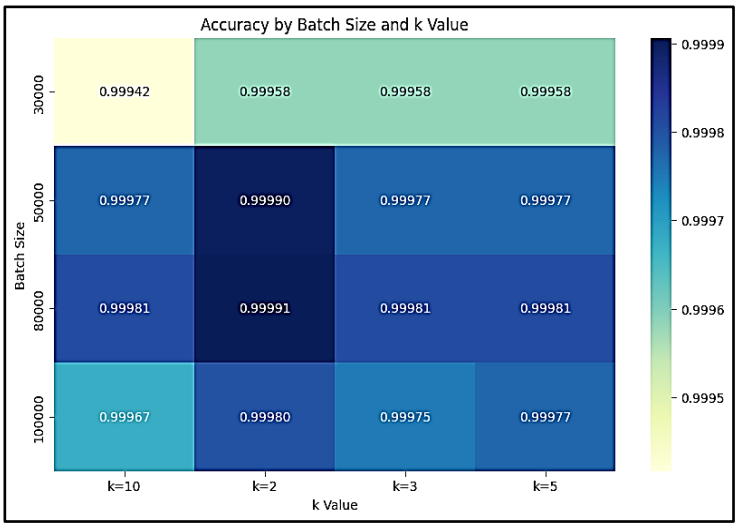

Figure 10: Accuracy with Different Batch Size and K Value.

Figure 11 shows further contributes intriguing insights into the influence of different K values on accuracy and training time across various sample sizes. Surprisingly, higher K values did not uniformly result in superior accuracy or longer training times. For instance, in a dataset with 100,000 samples, the K=2 configuration exhibited the highest accuracy but required the most training time. In contrast, the K=10 configuration demonstrated slightly lower accuracy but necessitated less training time. These findings challenge the assumption that increasing K leads to improved accuracy or prolonged training times across different sample sizes, emphasizing the need for a judicious selection of K based on dataset characteristics and performance trade-offs for optimal balance between accuracy and training efficiency.

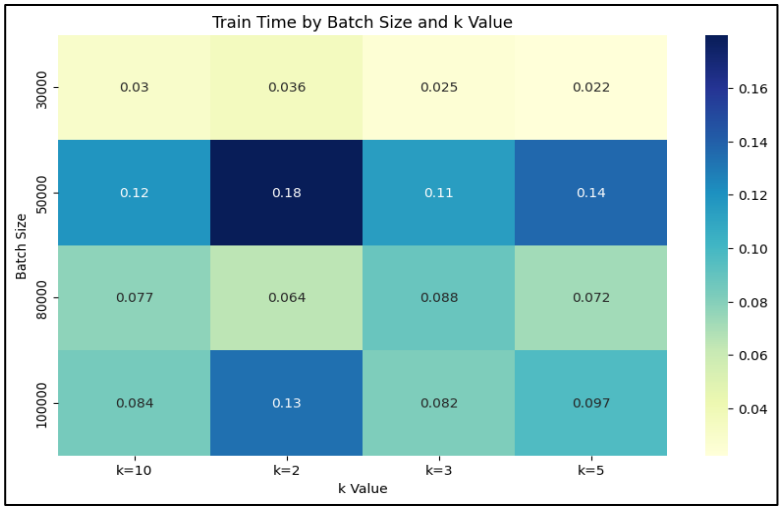

Figure 11: Training Time with Different Batch Size and K Value.

To summarize, the number of training samples and K in Fuzzy KNN interact complexly. More training samples provide more accurate but slower models. The optimum K seems problem-specific and does not follow the typical notion that bigger K values automatically result in more accurate models or longer training sessions.

5.5 Compared FuzzyKNN and Fuzzy Logic Results

In assessing the accuracy, flexibility, and noise resistance of Fuzzy Logic and Fuzzy KNN models:

- Accuracy: Fuzzy KNN consistently demonstrates superior accuracy, ranging from 99.95% to 99.97%, compared to Fuzzy Logic's 86%. highlights Fuzzy KNN's enhanced performance in accuracy.
- Flexibility: Fuzzy Logic excels in handling ambiguity and vagueness in complex real-world data. Meanwhile, Fuzzy KNN, being instance-based, adapts well to new training data, potentially leading to overfitting if not managed carefully.
- Noise Resistance: Fuzzy KNN's neighborhood approach enhances robustness to noisy input, protecting against outliers. Properly chosen K-values and the design of fuzzy membership functions help minimize noise, leveraging fuzzy logic's capacity to handle ambiguity.

6. Results and Discussion

The article discusses the honeypot system's model findings for malware detection. In a binary classification task and categorize occurrences as malware or normal. To explore how hyperparameters affect classification performance, our models are tested with different sample sizes (60,000, 100,000, and 160,000 examples) and K values (2, 3, 5, and 10).

6.1 Evaluation Metrics

These metrics can help evaluate the performance of the proposed honeypot system using PSO and fuzzy logic algorithms for detecting malware attackers and ensuring data privacy and information security. Confusion metrics: A confusion matrix is a table that summarizes a classification algorithm's performance. It presents the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions for each class. In a case, the matrix will have the following structure.

Accuracy: classifier accuracy. It is the percentage of correctly categorized cases, and the Precision: is the ratio of TP predictions to classifier positive predictions. It evaluates the classifier's malware detection, Recall (TP rate): the ratio of TP forecasts to positive cases. It evaluates the classifier's malware detection, the F1-score: PRE and REC harmonic mean. It combines PRE and REC metrics. The Sensitivity: REC, the ratio of TP predictions to positive cases. It evaluates the classifier's malware detection And Specificity: TN forecasts to total negative instances. It tests the classifier's usual case detection.

6.2 Libraries

The thesis extensively utilized various Python libraries to streamline machine learning (ML) processes in the development and optimization of models for the honeypot system. For preprocessing tasks, the combination of Numpy, Pandas, and Scikit-learn (sklearn) proved invaluable. Numpy, a scientific Python library, facilitated the manipulation of extensive multidimensional arrays and matrices, providing essential mathematical functions. Pandas, a versatile data manipulation toolkit, introduced DataFrame and Series structures, addressing tasks such as cleaning, modification, and analysis of data, including handling missing data and restructuring. Scikit-learn played a pivotal role in ML with its simplified data mining and analysis tools, offering preprocessing capabilities like scaling, encoding, and feature selection.

In terms of visualization, Matplotlib and Seaborn were the chosen tools. Matplotlib, a widely-used 2D charting library, supported static, interactive, and animated presentations with an object-oriented chart API. Seaborn, built on Matplotlib, specialized in statistical data visualization, providing a high-level interface for creating appealing and informative statistical visualizations.

For data optimization, Zoofs, a Python module with metaheuristic feature selection optimization, including Particle Swarm Optimization (PSO), played a crucial role. Theoptimization approach reduced dataset characteristics, improving both model performance and computational efficiency.

Table 1 encapsulates the outcomes derived from each model across diverse data shapes and K values, providing a detailed analysis of performance metrics such as ACC, error rate, PRE, REC, F1-score, sensitivity, and specificity. The comprehensive summary below presents a comparative overview of model performances, facilitating a nuanced understanding of accuracy, precision, recall, and overall effectiveness under various configurations and scenarios.

Table 1: The FuzzyKNN classification report.

Data shape	K	ACC	Error rate	PRE	REC	F1-score	Sensitivity	Specificity
60000	2	99.97%	0.0003	99.97%	99.97%	99.97%	99.97%	99.97%
	3	99.95%	0.0005	99.95%	99.95%	99.98%	99.97%	99.95%
	5	99.95%	0.0007	99.96%	99.95%	99.96%	100%	99.99%
	10	99.96%	0.0008	99.96%	99.96%	99.97%	99.97%	99.99%
100000	2	99.96%	0.0002	99.96%	100%	99.97%	100%	99.98%
	3	99.97%	0.0001	99.97%	99.97%	99.97%	100%	99.97%
	5	99.95%	0.00049	99.98%	100%	99.98%	100%	99.98%
	10	99.97%	0.00098	99.97%	99.97%	99.97%	99.96%	99.98%
160000	2	99.97%	0.00028	99.97%	99.97%	99.97%	99.98%	99.96%
	3	99.96%	0.00028	99.96%	99.96%	99.96%	99.97%	99.97%
	5	99.97%	0.00024	99.97%	99.97%	99.97%	99.96%	99.98%
	10	99.97%	0.00021	99.97%	99.97%	99.97%	99.96%	99.98%
200000	2	99.95%	0.00019	99.95%	100%	99.95%	99.97%	99.96%
	3	99.95%	0.00024	99.95%	99.95%	99.95%	99.96%	99.94%
	5	99.96%	0.00022	100%	100%	100%	99.96%	99.98%
	10	99.96%	0.00032	100%	99.96%	100%	99.95%	99.97%

Across varied data forms and K values, the model metrics exhibit relatively marginal variances, as illustrated in Table 2. Notably, model performance remains stable within the range of 60,000 to 160,000 instances, indicating FuzzyKNN's capability to handle larger datasets without notable degradation. The influence of K values on model performance is minimal, with models demonstrating consistent effectiveness across different K configurations. In a dataset of 50,000 cases, K = 3 displayed slightly higher ACC and specificity compared to K = 2, 3, and 10. The comparison with other methods, including Random Forest (RF), Decision Tree (DT), and XGBoost, showcases FuzzyKNN's superior accuracy, recall, F1-score, sensitivity, and specificity. FuzzyKNN outperformed with 99.97 percent accuracy and 0.0003 error rate, emphasizing its reliability and efficacy in data categorization.

6.3 Discussion

The innovative approach of combining Practical Swarm Optimization (PSO) with a fuzzy K-Nearest Neighbors (fuzzyKNN) model in a honeypot environment for malware detection presents a promising paradigm for enhancing cybersecurity. The utilization of PSO, inspired by social creatures' coordinated actions, introduces a dynamic and adaptive optimization strategy. The swarm's collective intelligence, guided by personal and collective knowledge, demonstrates efficacy in navigating complex solution landscapes.The synergy with the

fuzzyKNN model leverages the strengths of fuzzy logic and K-Nearest Neighbors, offering a robust framework for handling uncertainties inherent in malware patterns. Fuzzy logic enables a nuanced representation of uncertainties, while K-Nearest Neighbors facilitates data-driven classification based on similarity metrics. The hybridization of these techniques enhances the model's ability to discern subtle variations in malware behaviors and adapt to evolving threats.

Deploying the approach within a honeypot environment is particularly significant. Honeypots simulate real network conditions, attracting and detecting malicious activities. The combined PSO and fuzzyKNN model's adaptability and precision make it well-suited for navigating the complexities of diverse malware behaviors encountered within such environments.The innovative amalgamation not only showcases the potential for improving malware detection accuracy but also underscores the importance of leveraging nature-inspired optimization techniques in conjunction with advanced machine learning models. The discussion emphasizes the applicability of the approach in realistic cybersecurity scenarios, contributing to the ongoing efforts to fortify systems against evolving cyber threats.

7. Conclusion

Innovative Malware Detection: Practical Swarm Optimization and fuzzyKNN Model in Honeypot Environment presents a novel approach to combat the ever-evolving landscape of cyber threats. By leveraging the power of Practical Swarm Optimization (PSO) and the fuzzy k-Nearest Neighbor (fuzzyKNN) model in a honeypot environment, the system provides an efficient, accurate, and adaptive solution for detecting and analyzing malware. The combination of PSO and fuzzyKNN allows the model to optimize feature selection, handle uncertainty, and enhance the accuracy of malware classification.

The integration of honeypots into the malware detection system offers a safe and controlled environment for capturing real-world malware samples. The collected data from honeypots allows the model to continuously learn and adapt to emerging threats, making it a powerful defense against sophisticated cyberattacks. The PSO-fuzzyKNN model architecture, along with its training on honeypot data, ensures that the system is well-equipped to handle various malware families, including zero-day threats.

8. Recommendations for Future Research

As the field of cybersecurity continues to evolve, several areas of future research can further enhance the capabilities of the PSO-fuzzyKNN malware detection system:

1. Dynamic Feature Selection: Investigate dynamic feature selection techniques that allow the model to adaptively adjust the feature set based on the evolving characteristics of malware. Could improve the system's ability to handle new types of attacks and changes in attacker behavior.

2. Advanced Honeypot Technologies: Explore the use of advanced honeypot technologies, such as high-interaction honeypots with improved deception techniques, to attract more sophisticated attackers and capture their activities effectively.

3. Ensemble Techniques: Investigate the combination of the PSO-fuzzyKNN model with other machine learning algorithms or ensemble techniques to further enhance detection accuracy and robustness.

4. Explainable AI for Malware Analysis: Develop explainable AI techniques that provide insights into how the PSO-fuzzyKNN model arrives at its classification decisions. Could enhance the model's transparency and trustworthiness.

References

[1] Srilatha, D., & Shyam, G. K. (2021). Cloud-based intrusion detection using kernel fuzzy clustering and optimal type-2 fuzzy neural network. *Cluster Computing*, *24*(3), 2657-2672.

[2] M. A. Fauzi, A. T. Hanuranto and C. Setianingsih, "Intrusion Detection System using Genetic Algorithm and K-NN Algorithm on Dos Attack," *2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS)*, Manado, Indonesia, 2020, pp. 1-6, doi: 10.1109/ICORIS50180.2020.9320822.

[3] M. N. Al-Andoli, K. S. Sim, S. C. Tan, P. Y. Goh and C. P. Lim, "An Ensemble-Based Parallel Deep Learning Classifier With PSO-BP Optimization for Malware Detection," in *IEEE Access*, vol. 11, pp. 76330-76346, 2023, doi: 10.1109/ACCESS.2023.3296789.

[4] Abbasi, M. S., Al-Sahaf, H., & Welch, I. (2020). Particle swarm optimization: A wrapper-based feature selection method for ransomware detection and classification. In *Applications of Evolutionary Computation: 23rd European Conference, EvoApplications 2020, Held as Part of EvoStar 2020, Seville, Spain, April 15–17, 2020, Proceedings 23* (pp. 181-196). Springer International Publishing.

[5] Moodi, M., Ghazvini, M., Moodi, H., & Ghavami, B. (2020). A smart adaptive particle swarm optimization–support vector machine: android botnet detection application. *The Journal of Supercomputing*, *76*, 9854-9881. DOI: https://doi.org/10.1007/s11227-020-03233-x

[6] Hancitor malspam and infection traffic, 2019-02-05. [Online]. Available: https://isc.sans.edu/forums/diary/Hancitor+malspam+and+infection+traffic+from+Tuesday+20190205/24616/

[7] B. Kolosnjaji, G. Eraisha, G. Webster, A. Zarras and C. Eckert, "Empowering convolutional networks for malware classification and analysis," *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, 2017, pp. 3838-3845, doi: 10.1109/IJCNN.2017.7966340.

[8] A. Atapour-Abarghouei, S. Bonner and A. S. McGough, "A King's Ransom for Encryption: Ransomware Classification using Augmented One-Shot Learning and Bayesian Approximation," *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 2019, pp. 1601-1606, doi: 10.1109/BigData47090.2019.9005540.

[9] Al-Rimy, B. A. S., Maarof, M. A., Alazab, M., Shaid, S. Z. M., Ghaleb, F. A., Almalawi, A., ... & Al-Hadhrami, T. (2021). Redundancy coefficient gradual up-weighting-based mutual information feature selection technique for crypto-ransomware early detection. *Future Generation Computer Systems*, *115*, 641-658.

[10] Gill, S. S., Xu, M., Ottaviani, C., Patros, P., Bahsoon, R., Shaghaghi, A., ... & Uhlig, S. (2022). AI for next generation computing: Emerging trends and future directions. *Internet of Things*, *19*, 100514.

[11] Gazet, A. Comparative analysis of various ransomware virii. J. Comput. Virol. 2010, 6, 77–90. DOI:https://doi.org/10.1007/s11416-008-0092-2

[12] S. Baek, Y. Jung, D. Mohaisen, S. Lee and D. Nyang, "SSD-Assisted Ransomware Detection and Data Recovery Techniques," in *IEEE Transactions on Computers*, vol. 70, no. 10, pp. 1762-1776, 1 Oct. 2021, doi: 10.1109/TC.2020.3011214.

[13] Trans. Comput. 2020, 70, 1762–1776. [CrossRef]
Comito, C.; Forestiero, A.; Pizzuti, C. Word embedding based clustering to detect topics in social media. In Proceedings of the 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Thessaloniki, Greece, 14–17 October 2019; pp. 192–199.DOI: https://doi.org/10.1145/3350546.3352518

[14] Mehta, R., Mehta, S., & Sharma, S. (2021). A novel swarm intelligence-based approach for malware detection. In Proceedings of the 13th International Conference on Swarm Intelligence (pp. 99-111). Springer.

[15] Seyfari, Y., & Meimandi, A. (2023). A new approach to android malware detection using fuzzy logic-based simulated annealing and feature selection. *Multimedia Tools and Applications*, 1-25.

[16] R. M. Sharma and C. P. Agrawal, "A BPSO and Deep Learning Based Hybrid Approach for Android Feature Selection and Malware Detection," *2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT)*, Indore, India, 2022, pp. 628-634, doi: 10.1109/CSNT54456.2022.9787671.

[17] Potula, S. R., Selvanambi, R., Karuppiah, M., & Pelusi, D. (2023). Artificial Intelligence-Based Cyber Security Applications. In *Artificial Intelligence and Cyber Security in Industry 4.0* (pp. 343-373). Singapore: Springer Nature Singapore.

[18] Yang, X. (2023). Feature extraction algorithm of anti-jamming cyclic frequency of electronic communication signal. *Journal of Intelligent Systems*, *32*(1), 20220295. https://doi.org/10.1515/jisys-2022-0295

[19] Dabas, N., Ahlawat, P., & Sharma, P. (2023). An effective malware detection method using hybrid feature selection and machine learning algorithms. *Arabian Journal for Science and Engineering*, *48*(8), 9749-9767. https://doi.org/10.1007/s13369-022-07309-z

[20] Zhang, J., Yang, G., & Zhang, G. (2022). Fuzzy deep k-nearest neighbor classifier for malware detection. Neural Computing and Applications, 34(5), 1405-1415.

[21] Arunkumar, M., & Kumar, K. A. (2023). GOSVM: Gannet optimization based support vector machine for malicious attack detection in cloud environment. *International*

Journal of Information Technology, *15*(3), 1653-1660.. https://doi.org/10.1007/s41870-023-01192-z

[22] Biskup, J. (2022). 6.1 Keynote: Construction of Inference-Proof Agent Interactions. *Machine Learning under Resource Constraints-Applications*, 391.

[23] Ahmad, J., Shah, S. A., Latif, S., Ahmed, F., Zou, Z., & Pitropakis, N. (2022). DRaNN_PSO: A deep random neural network with particle swarm optimization for intrusion detection in the industrial internet of things. *Journal of King Saud University-Computer and Information Sciences*, *34*(10), 8112-8121. https://doi.org/10.1016/j.jksuci.2022.07.023

[24] Symantec Threat Intelligence Report. (2021).[Online]. Available: https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-27-2021-en.pdf

[25] Al-rimy, B.A.S.; Maarof, M.A.; Shaid, S.Z.M. Ransomware threat sucCheck Point Research. (2021). Check Point Research's 2021 Mid-Year Report. [Online]. Available: ttps://research.checkpoint.com/wp-content/uploads/2021/08/Check-Point-Research-2021-Mid-Year-Report.pdf

[26] Carbon Black. (2021). Modern Attacks on Memory Execution Prevention: Understanding and Combating Fileless and Memory Attacks. [Online]. Available: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/carbon-black/ebooks/modern-attacks-on-memory-execution-prevention.pdf

[27] Palo Alto Networks Unit 42. (2021). IoT Threat Report 2021. [Online]. Available: https://www.paloaltonetworks.com/resources/whitepapers/iot-threat-report-2021

[28] Symantec Threat Intelligence Report. (2022). [Online]. Available: https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-28-2022-en.pdf

[29] Hussain, A., Asif, M., Ahmad, M. B., Mahmood, T., & Raza, M. A. (2022, April). Malware detection using machine learning algorithms for windows platform. In *Proceedings of International Conference on Information Technology and Applications: ICITA 2021* (pp. 619-632). Singapore: Springer Nature Singapore.

[30] Hussain, Abrar, et al. "Malware detection using machine learning algorithms for windows platform." *Proceedings of International Conference on Information Technology and Applications: ICITA 2021*. Singapore: Springer Nature Singapore, 2022.https://doi.org/10.1007/978-981-16-7618-5_53

[31] Du, D., & Chang, C. K. (2018). Behavior-based malware detection using recurrent neural networks. In Proceedings of the International Conference on Information and Communications Security (pp. 342-357). Springer.

[32] da Costa, F. H., Medeiros, I., Menezes, T., da Silva, J. V., da Silva, I. L., Bonifácio, R., ... & Ribeiro, M. (2022). Exploring the use of static and dynamic analysis to improve the performance of the mining sandbox approach for android malware identification. *Journal of Systems and Software*, *183*, 111092. https://doi.org/10.1016/j.jss.2021.111092

[33] Alraizza, A., & Algarni, A. (2023). Ransomware detection using machine learning: A survey. *Big Data and Cognitive Computing*, *7*(3), 143. https://doi.org/10.3390/bdcc7030143

[34] Dorigo, M., & Stützle, T. (2004). Ant colony optimization. MIT Press.

[35] Reynolds, C. W. (1987, August). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (pp. 25-34).

[36] Hann, H., Nauditt, A., Zambrano-Bigiarini, M., Thurner, J., McNamara, I., & Ribbe, L. (2021). Combining satellite-based rainfall data with rainfall-runoff modelling to simulate low flows in a Southern Andean catchment. *Journal of Natural Resources and Development*, *11*, 01-19.. DOI: https://doi.org/10.18716/ojs/jnrd/2021.11.02

[37] Kumar, G., Singh, U. P., & Jain, S. (2022). An adaptive particle swarm optimization-based hybrid long short-term memory model for stock price time series forecasting. *Soft Computing*, *26*(22), 12115-12135.

[38] Zheng, J., Zhang, Z., Zou, J., Yang, S., Ou, J., & Hu, Y. (2022). A dynamic multi-objective particle swarm optimization algorithm based on adversarial decomposition and neighborhood evolution. *Swarm and Evolutionary Computation*, *69*, 100987. https://doi.org/10.1016/j.swevo.2021.100987

[39] Chen, B., Wen, G., He, X., Liu, X., Liu, H., & Cheng, S. (2023). Application of adaptive grid-based multi-objective particle swarm optimization algorithm for directional drilling trajectory design. *Geoenergy Science and Engineering*, *222*, 211431. https://doi.org/10.1016/j.geoen.2023.211431

[40] Jothi, C. S., & Belinda, C. M. (2023). Enhanced Fuzzy Logic Pre-Processing Technique Using Hybridized Bat and Particle Swarm Optimization Algorithm for Feature Selection. *International Journal of Intelligent Engineering & Systems*, *16*(3).

[41] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.

[42] Ali, H., Batool, K., Yousaf, M., Islam Satti, M., Naseer, S., Zahid, S., Gardezi, A. A., Shafiq, M., and Choi, J.-G. (2022). Security hardened and privacy preserved android malware detection using a fuzzy hash of reverse-engineered source code. Security & Communication Networks.

[43] I. Almomani *et al*., "Android Ransomware Detection Based on a Hybrid Evolutionary Approach in the Context of Highly Imbalanced Data," in *IEEE Access*, vol. 9, pp. 57674-57691, 2021, doi: 10.1109/ACCESS.2021.3071450.

[44] El-Ghamry, A., Gaber, T., Mohammed, K. K., and Hassanien, A. E. (2023). Optimized and efficient image-based iot malware detection method. Electronics, 12(3):708. https://doi.org/10.3390/electronics12030708

[45] M. P. Novaes, L. F. Carvalho, J. Lloret and M. L. Proença, "Long Short-Term Memory and Fuzzy Logic for Anomaly Detection and Mitigation in Software-Defined Network

Environment," in *IEEE Access*, vol. 8, pp. 83765-83781, 2020, doi: 10.1109/ACCESS.2020.2992044.

[46] Jäger, T., Mokos, A., Prasianakis, N. I., & Leyer, S. (2022). first_page settings Order Article Reprints Open AccessArticle Pore-Level Multiphase Simulations of Realistic Distillation Membranes for Water Desalination. *Membranes*..

[47] Membranes.Abraham, S. and Chengalur-Smith, I. (2010). An overview of social engineering malware: Trends, tactics, and im- plications. Technology in Society, 32(3):183–196. https://doi.org/10.1016/j.techsoc.2010.07.001

[48] Ali, W. (2019). Hybrid intelligent android malware detection using evolving support vector machine based on genetic algorithm and particle swarm optimization. *IJCSNS*, *19*(9), 15.

[49] Hu, H., Shan, H., Wang, C., Sun, T., Zhen, X., Yang, K., ... & Quek, T. Q. (2020). Video surveillance on mobile edge networks—a reinforcement-learning-based approach. *IEEE Internet of Things Journal*, *7*(6), 4746-4760.

[50] Wu, Y., Zhang, Q., Hu, Y., Sun-Woo, K., Zhang, X., Zhu, H., & Li, S. (2022). Novel binary logistic regression model based on feature transformation of XGBoost for type 2 Diabetes Mellitus prediction in healthcare systems. *Future Generation Computer Systems*, *129*, 1-12.

[51] Sharma, H., & Kumar, N. (2023). Deep learning based physical layer security for terrestrial communications in 5G and beyond networks: A survey. *Physical Communication*, 102002.

[52] Nivetha, S., & Inbarani, H. H. (2023). Novel Architecture for Image Classification Based on Rough Set. *International Journal of Service Science, Management, Engineering, and Technology (IJSSMET)*, *14*(1), 1-38.