Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-18180-5

Fuzzy K-Means with M-KMP: a security framework in pyspark environment for intrusion detection

Gousiya Begum[1,2] · S. Zahoor Ul Huq[3] · A. P. Siva Kumar[4]

Received: 25 January 2023 / Revised: 6 October 2023 / Accepted: 5 January 2024

Abstract

In recent times, IDS (Intrusion Detection System) has become a significant tool for improvising network security through the detection of abnormal and normal data. It is vital as it permits one to identify and respond to incoming malicious traffic. The intruders have also enhanced the inclusion of attacks in systems with a recent increase in data. Concurrently, ML (Machine Learning) algorithms can learn from corresponding data that has been afforded. With the provision of new data, the accuracy and efficacy of the ML model to take decisions to enhance with training. However, with the evolution of big data, ML has turned incapable of handling huge data interpretation issues which made most of the conventional systems explore high FP (False Positive) rates and low accuracy rates. This gave rise to pyspark which serves as a platform for addressing these issues that the ML method fails to solve. ML in pyspark is a scale and easy to use. Considering this, the present research intends to propose ML-based algorithms for classifying intrusion detection in a pyspark environment. This study proposes a security framework named Fuzzy K-Means with M-KMP (Modified-Knuth Morris Pratt) wherein the clustering is accomplished by Fuzzy K-means which is capable of exploring data points that potentially relate to multiple clusters. Whereas, M-KMP achieves information matching on the clustered data for assessment of the information occurrence on the allocated threat data that will serve as an assistance for security developers in attack prevention. The efficiency of this proposed work is confirmed through the results.

Keywords Intrusion detection system · Machine learning · Big Data · Fuzzy K-Means with Modified-Knuth Morris Pratt · Pyspark

1 Introduction

Over the years, big data has grown exponentially in volume. The significance of big data is not restricted to the volume of data, instead, it lies in the way in which it is used. Through the analysis of diverse data from varied sources, one could explore

✉ Gousiya Begum
gousiiya@gmail.com

Extended author information available on the last page of the article

Published online: 13 February 2024

several applications including streamlining resource management, enhancing operational functions, optimizing product development, and driving growth opportunities and new revenue thereby permitting optimal decision-making. Thus, big data has become a big deal for several industries. Advancements in IoT (Internet of Things) have created huge enhancements in the information collected, maintained and analyzed by organizations. In addition, big data has aroused the probability of unlocking huge insights for all kinds of industries ranging from small to large. However, protecting the privacy of these data has turned into a challenge for the security developer, particularly with extensive use of the internet and prompt progress of data retrieved from multiple sources which create maximum space for the intruders to promote malicious attacks. Conventionally, intrusion detection methods have been constructed through signature-based strategies indicating that cyber-attacks are identified by signature matching (for instance: renowned bit streaming or event sequences). In this case, only the known attacks could be detected by using these methods. Additionally, as new attacks have been identified over time, new signatures should be constructed requesting human intervention. Detection throughput has been further minimized as numerous signatures have to be assessed [1]. Thus, detection could be performed only after the occurrence of a cyber-attack. Concurrently, an AI (Artificial Intelligence) based intrusion detection system tries to learn or benchmark the typical types of network traffic retrieved by the IoT devices. This assists in identifying the anomalies using methods and deviances from typical traffic kinds.

Several researchers exploited ML-based algorithms to produce an accurate intrusion detection system and reduce incorrect positive rates for intrusion detection. Conversely, the technique of machine learning acquires a lengthy time in analyzing and classifying the big data. Through machine learning and big data method applied for IDS, the challenges can be resolved by enhancing accuracy and reducing speed and computational time [2]. Procuring important information, analyzing the data, and interpreting relevant data can be accomplished by utilizing machine learning technology. However, the traditional machine learning method is incapable of successfully dealing with a large form of data interpretation problem. This gives rise to the introduction of spark architecture which forms a platform to address these issues that the machine learning method fails to solve [3]. Considering this, the study [4] has suggested a spark big data technique that deals with huge data in IDS that demands minimizing computation period and to achieve efficient classification. IDS classification method termed spark-chi-SVM has been recommended. The pre-processing method has been used initially to translate the categorical data to numeric form and at the moment, the dataset is standardized for the persistence of improving the ordering efficiency. Next, the method of chi-square selector has been applied to minimize the dimensionality of the dataset to enhance the grouping efficiency and decrease the computation time to follow up for the next stage. Finally, for data classification, SVM (Support Vector Machine) has been used. To resolve the optimization, SVM with SGD has been more specifically used. In addition, the comparison among LR (Logistic Regression) and SVM classifier on Apache spark have been performed based on AUROC (Area under the curve), time metrics, and AUPR (Area under Regression Recall) curve. For better performance, KDDCUP99 has also been tested which has explored better performance.

Nevertheless, for combating malware, even more, reliable computerized detection tools have been needed [5]. Several researchers have established automatic evaluation cognitive systems for handling malware that resist attacks. To create these automated tools, persistent evaluation of malware has been needed for updating detection tools along with the behavior and pattern of malware and existing malware variants. Considering this, the conventional study [6] has used the K-means methodology from ML libraries on the spark platform for assessing if the incoming values of the network are normal. Nearly, four hundred data have been utilized with data attained from KDD-CUP-1999 data. Through the use of the suggested method, ten abnormal behaviors have been identified. Similarly, the study [7] has used a clustering methodology relying on PMBKM (Principal Component Analysis Mini Batch K-Means) to perform intrusion detection. Considering intrusion detection datasets like KDDCUP99, the full dataset and 10% of the dataset have been tested. The dataset has been pre-processed and the PCA method has been used for minimizing the dimensions for enhancing the clustering effectiveness. A comparative evaluation with existing methods has revealed the better efficacy of the recommended system [8]. Though existing works have performed better, they have been lacking an accuracy rate and a high FP (False Positive) rate. To avert these pitfalls, the present study endeavors to propose a security framework based on ML to identify intrusions in big data sets.

The main contributions of this study are listed below,

- To perform clustering using the proposed Fuzzy K-means for procuring flexibility to explore the data points that potentially pertain to multiple clusters.
- To accomplish information matching on the clustered data through the use of proposed M-KMP (Modified-Knuth Morris Pratt) to effectively assess the information occurrence on the allocated threat data which assists in attack prevention.
- To evaluate the performance of the proposed system concerning the accuracy, recall, F1-score and precision for validating its effectiveness in detecting intrusions.

1.1 Paper organization

The manuscript is organized as follows, Section 2 discusses the conventional works along with the identification of problems, and Section 3 presents the proposed flow, Pseudocode/algorithm and all the relevant information relating to the proposed methods. Subsequently, the outcomes attained from the execution of the proposed system are discussed in Section 4 with the overall summary of the proposed work in Section 5.

2 Review of existing work

Several conventional works have aimed to perform intrusion detection with big data by exploiting various ML algorithms. These studies are reviewed in this section by emphasizing the drawbacks faced by them.

The study [9] has performed an extensive analysis of existing approaches for malicious social bots. Analyzed approaches include graph-based, ML and evolving methods. It has been emphasized that limitations exist by solely depending on graph-based or ML strategies. The subsequent issue is entangled with the detection of synchronized bots that rely on overlapping degrees and synchronicity levels. Moreover, it has also been explored that, an improved detection rate could be accomplished with the need for gathering huge features or employing computationally expensive techniques. Considering the significance of features, the research [10] has gathered all the accessible features in windows-performance monitors for determining the features that might be valuable for data-driven intrusion detection. Cyber-attack consequences have been simulated by including five attacks such as DoS (Denial of Service), data tampering, false data injection, MITM (Man-In-The-Middle) attack and data exfiltration. These have been considered to construct detection models. Classification models relying on the host system and network data have been studied that included KNN (K-Nearest Neighbour), RF (Random Forest), Bagging and DT (Decision Tree) for affording a secondary defense line for detecting cyber-attacks when the layer for intrusion prevention fails. Outcomes have recommended the better performance of all the considered models excluding RF. In this suggested system, AAKR (Auto-Associative Kernel Regression) framework has been researched to strengthen the early detection of an attack. Results have exposed that, the suggested system has been able to identify the physically influencing cyber-attacks before the occurrence of significant consequences. The main cause behind these attacks occurs by deceiving users with malign URLs (Uniform Resource Locators). Consequently, detecting malicious URLs seems to have gained significant attention in recent years. Accordingly, the article [11] has endorsed a method for identifying malign URLs through ML-based methods namely RF and SVM (Support Vector Machine). Besides, big data technology has also been used for enhancing detection capability. Empirical outcomes have exposed the better performance of RF. In addition, ELM (Extreme Learning Machine) has been suggested for the detection of domain names of the malware. Domain names have been classified by the features retrieved from several resources. Empirical outcomes have explored the satisfactory performance of the suggested method [12].

Moreover, the remote monitoring in this suggested study is implemented using the fuzzy inference for analysing the reliability of the detection map. When the map have been reliable, the target feature of each of the search area is transformed to a substitute template. The complete model is evaluated using multiple assessment metrics and robustness of the model is further improved [13].

Relevant characteristics of human internal thinking are combined in the suggested approach, by screening the movement information and target have been located using fuzzy thinking. The alternative selection strategy which have been based upon the thinking set is applied, providing the alternates among the location of thinking reasoning for further effective visual monitoring of the target upon OTB-2015 and UVA123 dataset have been carried out in the suggested approach [14].

Following this, the study [15] has specifically focussed to identify SQL injection attacks through ML. Few methods relied on dynamic-SQL query modeling. This has been compared with a collection of legal SQL query models. Though these methods have been easy to execute, possessing all such legal models have been complex. A deficiency of sufficient legal models could result in the maximum rate of false positives and hence might lead the web application out-of-service. Another division of methodologies relied on randomizing SQL queries. The main issue of these methods has been that the randomization method could alter the intended query of the user which might also restrict the probable input of users. Thus, different ML algorithms have been used for detecting and classifying malware through the use of features retrieved after static malware analysis and dynamic malware analysis. However, the article [16] has used a hybrid strategy to detect malware and classify it. Experimentation outcomes have indicated the better performance of the hybrid approach in comparison to cases when dynamic and static features have been regarded [17]. Similarly, LR (Logistic Regression), NN (Neural Network), DT and G-NB (Gaussian Naïve-Bayes) have been used for classifying intrusion detection [18]. Experimental results have revealed that 61%

have been normal attacks, while, 39% have been an anomaly. In the experimentation, 22,544 samples have been retrieved as training data for constructing training models to choose ML classifiers. Experimental outcomes have explored the better performance of DT in comparison to LR, NN, and G-NB with a minimum false negative rate. On contrary, G-NB has accomplished average detection rate, recall, and precision for identifying normal and anomaly packets. Likewise, the article [19] has considered an algorithmic model which uses social media BDA (Big Data Analytics) and statistical ML for cyber risk prediction [20, 21]. Initially, the data have been evaluated through descriptive analysis namely histogram, cluster dendrogram, pyramid analysis, and commonality and word cloud. Through this evaluation, words such as exploit, vulnerability, and apache are occurring frequently. Subsequently, the prediction has been accomplished through the use of K-NN (K-Nearest Neighbour), DT, ANN (Artificial Neural Network), SVM, and NB. Comparative evaluation has revealed the better accuracy of ANN. Furthermore, the study [22] has utilized different kinds of ML approaches for identifying intrusion. The performance and efficacy of several ML approaches have been validated by varied parameters. DT has attained less false negative value.

Considering this, the study [23] has suggested an intrusion detection system relying on DT. Comparison has been performed with K-NN and NB. Testing has been performed with full and 10% datasets. The suggested technique has not only detected four attack kinds, but it also permits the identification of 22 attack kinds. Empirical outcomes have explored the precise and better performance of the suggested system. The study intended to research intrusion detection for varied attack types in the future. Taking this into account, the research [24] has used a black box fuzzing method for detecting vulnerabilities due to parameter tampering and XQuery injection. XiParam has been developed and then tested on susceptible applications suggested with the native-XML database as the backend. Empirical assessment has revealed a better result of the prototype against the identification of vulnerabilities and XQuery injection [25]. To afford better identification for huge-scale intrusion detection, DBN (Deep Belief Network) and ensemble SVM have been used which possessed its reliance on Apache Spark [26–28]. From the attained outcomes, it has been found that the deep framework has made substantial development in feature learning for attack detection [29, 30]. Further, a methodology has been presented which gathers a user list with their corresponding followers sharing posts that have identical interests from hacker groups on Twitter. The list is constructed under a collection of recommended keywords that pertain to terms utilized by hackers in tweets. Following this, a complex network has been generated for individual users for determining associations within them about closeness, betweenness and network centrality. After the extraction of these values, a dataset with influential users especially in the hacker community has been assembled. Then, tweets associated with users in the extracted dataset have been collected and grouped into negative and positive classes. The outcome of such a process has been used with the ML process through the employment of varied algorithms [31, 32].

Similarly, the study [33] has suggested a model for the automatic extraction of terrorist-oriented attack information through Twitter and generating alerts. This has been followed by WMD (Word Mover's Distance) and fastText word representation for grouping semantically associated tweets that led to 2:5 times enhancement in clustering [34]. Recommended bLSTM has shown 96% prediction. To enhance the performance rate, the article [35] has presented a comprehensive analysis of SDA (Secure Data Analysis) through DL and ML models. Comparative evaluation of conventional models utilized for SDA has worked based on the attack kinds, parameters for attack mitigation and targeted area. Analytical results have explored that, when a method has performed well for attack detection, it might not perform similarly to identify other attack kinds. However, all such methods have not been executed for assessing the performance in confirming the reproducibility of results. Hence, further analysis has to be performed for determining the efficient method for intrusion detection (Table 1).

2.1 Problem identification

Substantial issues identified through the review of the above existing works are discussed in this section,

- For detecting unknown attacks through host-system and network data, unsupervised big data models have to be researched for enhancing the second line of defense [10].
- Other publicly accessible datasets have to be used with unsupervised classifiers [18].
- A collection of dynamic keywords that have been capable of integrating and eliminating new keywords relying on their impact on the hacker community has to be considered. Further, an alert framework has to be developed for primary attacks for notifying organizations and individuals regarding suspected users in twitter media [31].

3 Proposed methodology

The study intends to perform intrusion detection in big data sets using the proposed security framework. Though traditional works have endeavored to accomplish this, they showed a maximum FP rate and inaccurate detection. Considering these drawbacks, the present research aims to detect intrusions based on the flow as shown in Fig. 1. In this case, the dataset is initially taken as input which is then analyzed through word embedding which assists in capturing the similarities between words. This is followed by the allocation of pre-defined threats. Simultaneously, a security framework is proposed which encompasses a sequence of processes where the word embedding is considered and the K-Means algorithm is employed. Subsequently, normalization is performed by utilizing fuzzy string to attain the clustered data. After this, information is matched on clustered data through the proposed M-KMP algorithm. Then, information occurrences are checked on the allocated threat data. If information occurs, the threat is marked and it is updated in a pre-defined threat which will assist as a measure for preventing attacks. If the information doesn't occur on the allocated threat data, then iteration continues. Finally, the proposed system is assessed through performance metrics for confirming its efficacy.

Description of the framework From the input dataset, the data analysis are carried out using the two various approaches comprising, the file loading and word embedding. These are used in allocating the Pre-defined attacks to the networks. Followed by, the construction of the security framework encompasses the model “Fuzzy K-Means with Modified KMP algorithm”. Under this framework, the word embedding are proceeded using the K-means algorithm, and are normalised using the Fuzzy string approach. The

Table 1 State-of-art-approaches representing the aim and their respective outcomes

Reference Number	Aim of the study	Approach implied	Outcomes
[10]	Feature determination for data-driven intrusion detection.	DoS (Denial of Service), data tampering, false data injection, MITM (Man-In-The-Middle) attack and data exfiltration are evaluated	KNN (K-Nearest Neighbour), RF (Random Forest), Bagging and DT (Decision Tree) for affording a secondary defence line for detecting cyber-attacks when the layer for intrusion prevention fails
[12]	RF have been used for enhancing detection capability	ELM (Extreme Learning Machine) has been suggested for the detection of domain names of the malware	ELM is both effective and efficient to identify malicious domains and therefore enhance the current detection mechanism of APT attacks.
[17]	hybrid strategy to detect malware and classify intrusion	Dynamic and static features have been regarded using the hybrid approach. singular value decomposition technique for reducing dimensions of string features. Secondly, Shannon entropy is computed over the printable strings	The technique is validated with 16,489 malware and 8422 benign files. Our experimental results show the accuracy of 99.54% in malware detection using ensemble machine learning algorithms.
[23]	intrusion detection system relying on DT	Comparison has been performed with K-NN and NB.	The suggested model detected four attack kinds, but it also permits the identification of 22 attack kinds. Empirical outcomes have explored the precise and better performance of the suggested system
[22]	ML approaches for identifying intrusion. grealistic CSE-CIC IDS-2018 network datasets have been implied for the analysis.	ML algorithms have been implied to classify benign or malicious packets in UAV networks to enhance security	DT has attained less false negative value
[13]	fuzzy inference for analysing the reliability of the detection map	The tracking algorithm based on the Siamese network is implied for detection map. effects the probability that any position in the search area is the centre of the target's bounding box, and the maximum value of the detection map is the centre of the target's bounding box predicted by the algorithm	The target feature of each of the search area is transformed to a substitute template. The complete model is evaluated using multiple assessment metrics and robustness of the model is further improved

Table 1 (continued)

Reference Number	Aim of the study	Approach implied	Outcomes
[14].	Relevant characteristics of human internal thinking are combined	Screening the movement information and target have been located using fuzzy thinking. Moreover, lternative selection strategy upon the thinking set is applied, by providing the alternates among the location of thinking reasoning.	Integration of the proposed edge learning method with the IoT can be well applied to the construction of smart cities and future generation systems
[36]	DL based method for crop selection and in forecasting crop and in yield production forecasting.	Crop data sets can be used to classify soil fertility, crop selection, and many other aspects using ML algorithms. DL algorithms can be applied to farming data to do time series analysis and crop selection.	Suggestion of the appropriate crop recommendations using ML and DL techniques for crop yield by using time series analysis will aid in reducing the food insufficiency in the future.

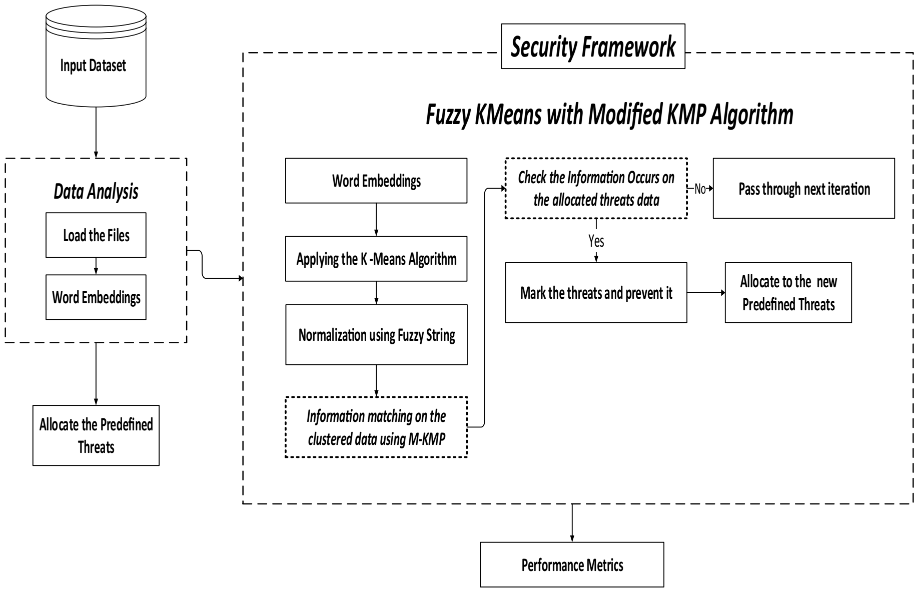
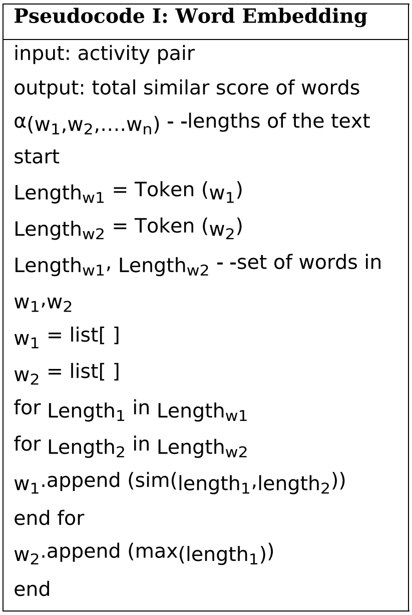

Input Dataset
Data Analysis
Load the Files
Word Embeddings
Allocate the Predefined Threats
Security Framework
Fuzzy KMeans with Modified KMP Algorithm
Word Embeddings
Applying the K -Means Algorithm
Normalization using Fuzzy String
Information matching on the clustered data using M-KMP
Check the Information Occurs on the allocated threats data
No
Pass through next iteration
Yes
Mark the threats and prevent it
Allocate to the new Predefined Threats
Performance Metrics

Fig. 1 Overall flow of the proposed system

information retrieved, which matches are clustered using the MKMP algorithm. This is followed for a complete check upon the threats to the data. If the loop passes, the threats are marked and are made to prevent further. If the data check did not pass, the loop moves to following same iteration for threat detection and prevention. Finally, the overall performance of the model is validated using the probabilistic performance metrics comprising Accuracy, Precision, Recall and the F1-score rates for evaluating the model performance of threat detection and prevention.

3.1 Word embedding

Informative representation of input could possess a massive influence on the overall performance of a model. As word embedding seems to be the optimal strategy to achieve this, the present study regards word embedding wherein phrases or words are mapped into real numbers in vector form. In the present study, word embedding is performed based on Pseudocode-I. A functionα is considered for computing similarity amongst a set of activities. Initially, the activity labels are tokenized for generating two-word sets. Then, vector representations of the label are generated for the word in the initial set and an individual word in the subsequent set. In this stage, a similarity score is computed wherein words with high similarity are integrated and the score corresponding to this pair is stored. The same process is reiterated for all the probable words in initial pairs. Finally, the maximum similarity scores for individual pairs are integrated for generating a unified similarity score.



```
Pseudocode I: Word Embedding
input: activity pair
output: total similar score of words
$\alpha(w_1,w_2,....w_n)$ - -lengths of the text
start
$Length_{w1}$ = Token ($w_1$)
$Length_{w2}$ = Token ($w_2$)
$Length_{w1}$, $Length_{w2}$ - -set of words in
$w_1$,$w_2$
$w_1$ = list[ ]
$w_2$ = list[ ]
for $Length_1$ in $Length_{w1}$
    for $Length_2$ in $Length_{w2}$
        $w_1$.append (sim($length_1$,$length_2$))
    end for
    $w_2$.append (max($length_1$))
end
```

This process works based on the construction of the meaning of individual words that correspond with their neighboring words to attain clusters as shown in Fig. 2.

3.2 Fuzzy K-Means

Fuzzy K-Means is a simple algorithm for grouping data and performing clustering by portioning the samples into a set with less cluster error. The overall algorithm of Fuzzy K-Means is shown in Algorithm-I. In this algorithm, three major stages are involved. Initially, the cluster centroids are chosen. Then, individual samples are assigned to the neighboring centroid and at last, the centroids are updated for an individual cluster. This algorithm typically adopts Euclidean and cosine distance.

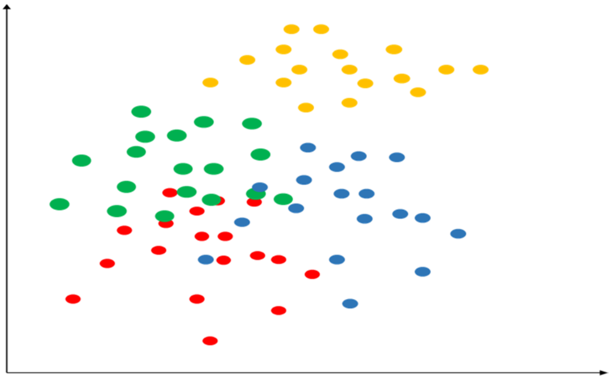


Fig. 2 Word embedding

Algorithm 1 Fuzzy K-Means

```
Step 1: Initial cluster centroid= $c_1, c_2, \ldots\ldots c_p$
selected samples =$s_1, s_2, \ldots\ldots s_k$
Step 2: The similarities between each sample and centroid
are Computed and each sample is assigned to the nearest
 centroid
Step 3: The mean of the sample in each cluster is calculated
as a new cluster.
Step 4: k-means usually adopts Euclidean distance and
cosine distance.
Euclidean distance:
 $Dis(s_i, c_i) = \sqrt{\sum_{l=1}^{V} (s_{il} - c_{jl})^2}$     $i = 1, \ldots\ldots M; j = 1 \ldots\ldots p$
Cosine distance:
 $Dis(s_i, c_i) = \frac{s_i^T c_j}{|s_i||c_j|}$     $i = 1 \ldots M; j = 1 \ldots\ldots p$
```

The distance between the adopted centroid and the sample directly impacts the clustering outcomes and the last centroids will possess less distance. The clusters attained after employing Fuzzy K-Means are shown in Fig. 3.

The main merit of the proposed Fuzzy K-Means is that it permits the gradual data point membership for clusters computed in (1, 2). This affords flexibility for expressing data points that pertain to multiple clusters as shown in Fig. 4.

3.3 Modified-KMP (Knuth Morris Pratt) matching

The present study considers KMP which uses a pre-processed pattern for attaining optimal outcomes. This algorithm is identical to the naïve matching algorithm. It replicated shifts in order from (1 to TL – PL) thereby summarizing if the specific arrangement matches with the shift. Modification is that this algorithm makes use of facts gathered from fractional and text matching for avoiding over-shifts which are certain to not impact a match and the overall process is shown in Pseudocode-II.

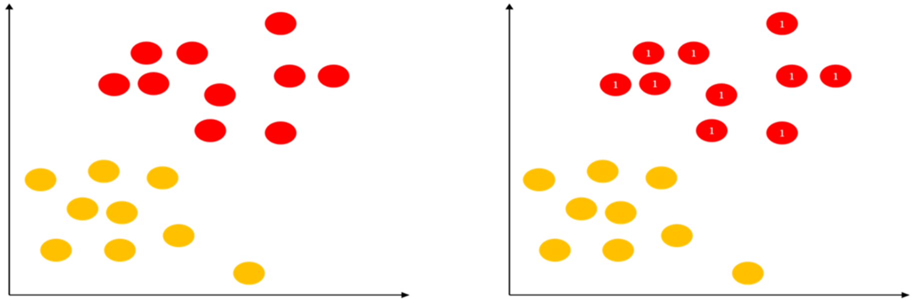


Fig. 3 Clustering after employing Fuzzy K-Means

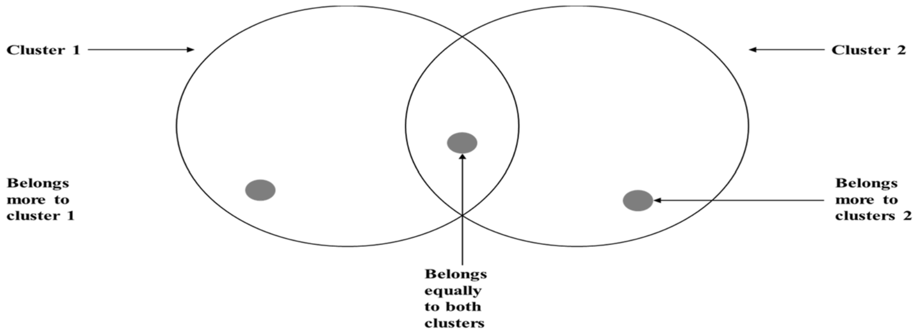

Cluster 1
Cluster 2
Belongs more to cluster 1
Belongs more to clusters 2
Belongs equally to both clusters

Fig. 4 Clustering data (words) by Fuzzy K-Mean belonging to multiple clusters

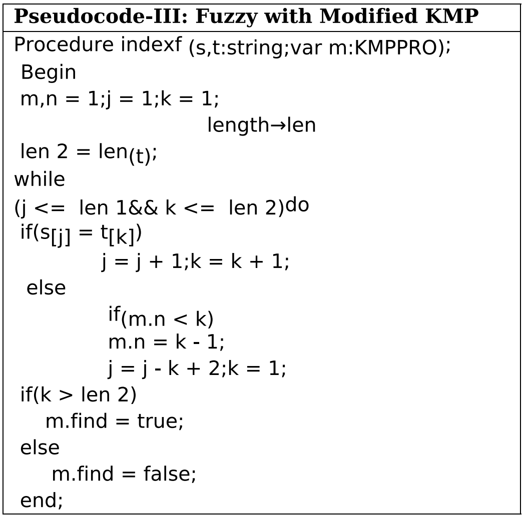
3.4 Fuzzy with Modified KMP

The fuzzy similarity indicates the proportion of the number of similar characters encompassed in two strings to the overall characters in the shortest string. In this case, the ratio is utilized for representing fuzzy similarity. This fuzzy is considered with Modified KMP and the overall Pseudocode is shown in Pseudocode-III. The function header encompasses three parameters such as S, T and M where S, T indicates the string type, while, M represents the KMPPRO that is exploited for finding if the considered strings

Pseudocode-II: Modified-KMP

```
KMP Search (P, T)
Step 1:PL→Pattern
 TL →Text
Step 2: int C[]
Step 3: j=0
Step 4: matching array (PAT, M, C)
Step 5: i=0
Step 6: While (i < TL)
Step 7: if (pat[j] = =txt [i])
Step8: j++
Step 9: i + +
Step 10: if (j= = PL)
Step 11: printf (“found pattern at index %d”, i-j)
Step 12:j = c[j - 1]
Step 13: elseif(i < TL&&pat[j]! = txt[i])
Step 14: if( j! = 0)
Step 15: j = c[j - 1];
Step 16: else
Step 17:i = i + 1
```

have exact match and similarity. An assumption has been made claiming that character pointers that have to compare the main string (S) and pattern string (T) are j and k. The initial values of these strings are considered as 1. In this matching process, when $s[j] = t[k]$, j and k values are enhanced by 1. Otherwise, values of m and j are compared. When m.n < k, then m.n is assigned for k. Otherwise, the value remains unchanged. Then, $[j]$ retreats to $j - k + 2$, while, k is reassigned as 1. This process continues until the j value is more than the main string length and the k value exceeds the pattern string value. In this way, the *j* values could be utilized for judging if two of the strings match. When two strings don't match, maximum similarity matches are recorded. The Pseudocode of Fuzzy with Modified KMP is shown in Pseudocode-III.



Pseudocode-III: Fuzzy with Modified KMP

```
Procedure indexf (s,t:string;var m:KMPPRO);
 Begin
 m,n = 1;j = 1;k = 1;
                    length→len
 len 2 = len(t);
while
(j <=  len 1&& k <=  len 2)do
 if(s[j] = t[k])
          j = j + 1;k = k + 1;
  else
           if(m.n < k)
           m.n = k - 1;
           j = j - k + 2;k = 1;
 if(k > len 2)
    m.find = true;
 else
     m.find = false;
 end;
```

3.5 Fuzzy K-Mean with Modified KMP

The study proposes a security framework named Fuzzy K-Mean with Modified KMP where the KMP matching approach utilizes degenerating property of the respective pattern. In this case, degenerating property indicates the similar sub-patterns that are appearing several times in a pattern. The main idea of this algorithm is that, when the algorithm identifies a mismatch succeeding certain matches, few text characters seem to be already known. This information is taken as a merit for avoiding matching characters that are already known to match anyway. The overall process is projected in Pseudocode-IV.

As shown in Pseudocode-IV, the dataset is taken as input to detect the threats and prevent them based on certain processes where the pre-defined threats are allocated. Then, the activity word pairs are initialized. Subsequently, the dataset is taken as input and word embedding is performed. Following this, similar words are clustered from word embedding and assigned an individual item to the cluster that has the closest

Pseudocode-IV: Fuzzy K-Mean with Modified KMP

```
INPUT: DATASET
OUTPUT: MARK THE THREATS AND PREVENT IT
begin
"Allocated predefined threats"
anonymous
 =  "fsck || dfsadmin || FileSystem|| balancer|| expunge||chgrp||cacheadmi
initialize activity pair of words
input dataset S = $s_1,s_2,......s_k$
$\alpha(w_1,w_2,....w_n)$ - -lengths of the text in the dataset
start
$Length_{w1}$, $Length_{w2}$ - set of words in $w_1,w_2$
$w_1$ = list[ ]
$w_2$ = list[ ]
$word_{emb}$ = total similar score of words
cluster the similar words from the $word_{emb}$
C =  $c_1,c_2,.......c_p$
$word_{emb}$ = total similar score of words
assign initial values
repeat
assign each item $s_i$ to the cluster which has the closest mean
calculate the mean cluster for each cluster
$Dis(s_i,c_i) = \sqrt{\sum_{l=1}^{v} (s_{il} - c_{jl})^2}$
until criteria are met
Allocated predefined threats
PL =  Search ( "fsck || dfsadmin || FileSystem|| balancer|| expunge||chgrp||
PL→Pattern
Word = " dfsadmin"
$w_1$→Text
  int C[ ]
j=0
Matching array (PL , C)
for  i = 0, i++
While (i < $w_1$)
if (pat[j] = =txt [i])
j++
printf ("found pattern at index ")
Procedure indexf($word_{emb}$, PL, Allocated predefined threats)
Begin
$Length_{w1}$ = Token ($w_1$)
$Length_{w2}$ = Token ($w_2$)
while
(j <=  $Length_{w1}$ && k <=  $Length_{w2}$)do
 if($s_{[j]}$ = $t_{[k]}$)
             j = j + 1;k = k + 1;
  else
              if(m.n < k)
              m.n = k - 1;

            j = j - k + 2;k = 1;
 if(k > $Length_{w2}$)
    m.find = true;
 else
     m.find = false;
 end;
```

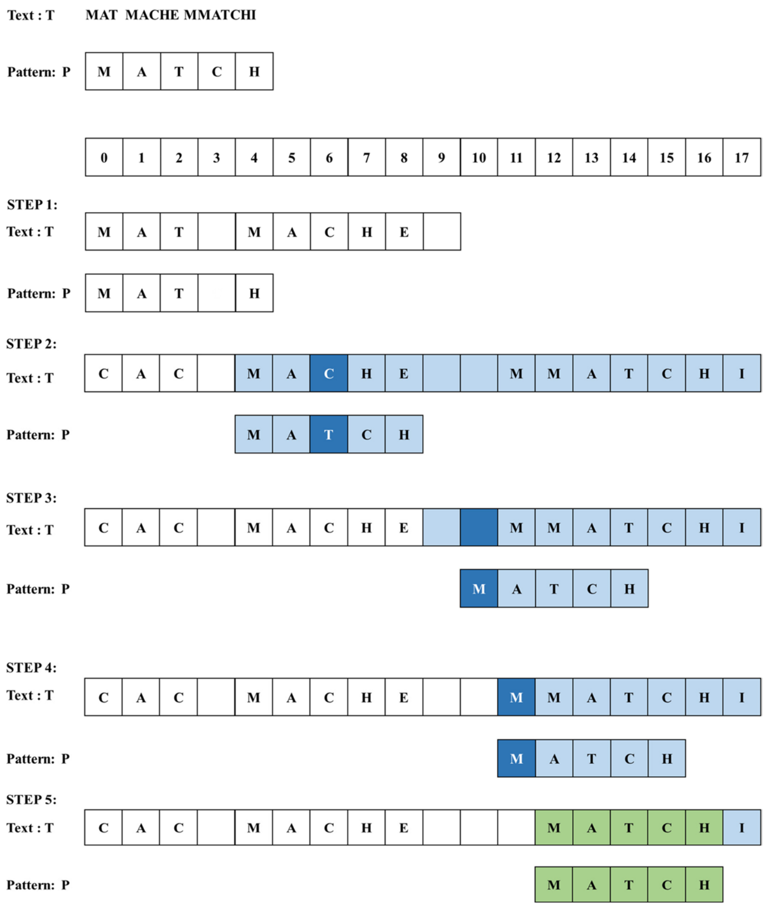
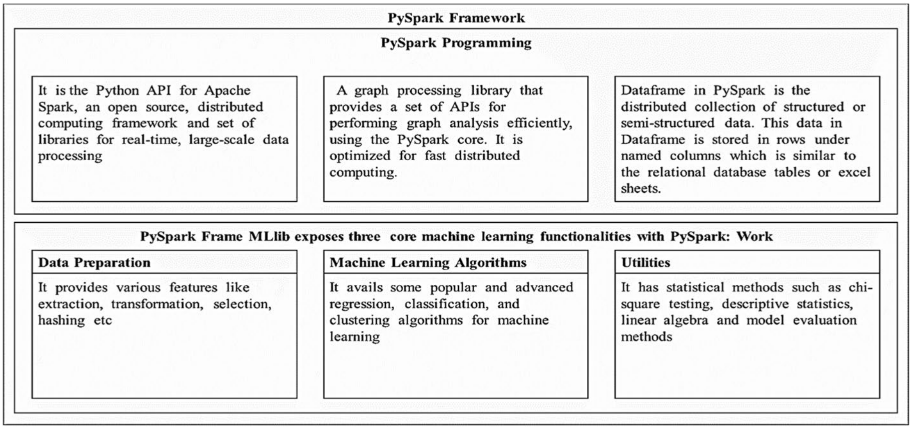


Fig. 5 Pattern matching



PySpark Framework

PySpark Programming

It is the Python API for Apache Spark, an open source, distributed computing framework and set of libraries for real-time, large-scale data processing

A graph processing library that provides a set of APIs for performing graph analysis efficiently, using the PySpark core. It is optimized for fast distributed computing.

Dataframe in PySpark is the distributed collection of structured or semi-structured data. This data in Dataframe is stored in rows under named columns which is similar to the relational database tables or excel sheets.

PySpark Frame MLlib exposes three core machine learning functionalities with PySpark: Work

Data Preparation

It provides various features like extraction, transformation, selection, hashing etc

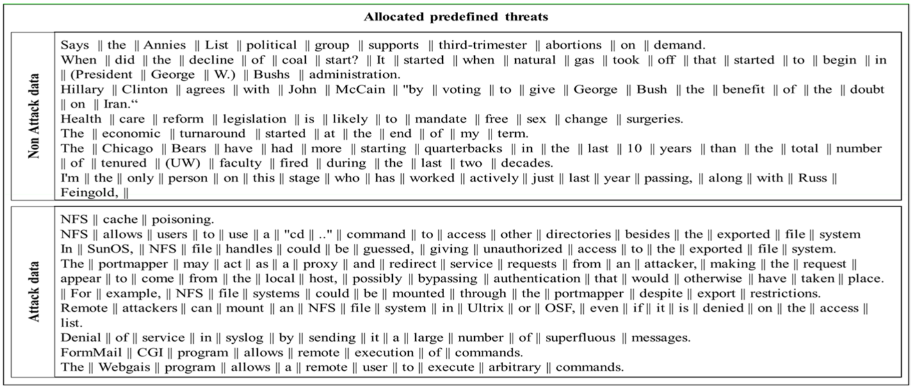
Machine Learning Algorithms

It avails some popular and advanced regression, classification, and clustering algorithms for machine learning

Utilities

It has statistical methods such as chi-square testing, descriptive statistics, linear algebra and model evaluation methods

Fig. 6 Pyspark framework



<table>
<tr><th colspan="2">Allocated predefined threats</th></tr>
<tr><td>Non Attack data</td><td>Says || the || Annies || List || political || group || supports || third-trimester || abortions || on || demand.
When || did || the || decline || of || coal || start? || It || started || when || natural || gas || took || off || that || started || to || begin || in || (President || George || W.) || Bushs || administration.
Hillary || Clinton || agrees || with || John || McCain || "by || voting || to || give || George || Bush || the || benefit || of || the || doubt || on || Iran.“
Health || care || reform || legislation || is || likely || to || mandate || free || sex || change || surgeries.
The || economic || turnaround || started || at || the || end || of || my || term.
The || Chicago || Bears || have || had || more || starting || quarterbacks || in || the || last || 10 || years || than || the || total || number || of || tenured || (UW) || faculty || fired || during || the || last || two || decades.
I'm || the || only || person || on || this || stage || who || has || worked || actively || just || last || year || passing, || along || with || Russ || Feingold, ||</td></tr>
<tr><td>Attack data</td><td>NFS || cache || poisoning.
NFS || allows || users || to || use || a || "cd || .." || command || to || access || other || directories || besides || the || exported || file || system
In || SunOS, || NFS || file || handles || could || be || guessed, || giving || unauthorized || access || to || the || exported || file || system.
The || portmapper || may || act || as || a || proxy || and || redirect || service || requests || from || an || attacker, || making || the || request || appear || to || come || from || the || local || host, || possibly || bypassing || authentication || that || would || otherwise || have || taken || place.
|| For || example, || NFS || file || systems || could || be || mounted || through || the || portmapper || despite || export || restrictions.
Remote || attackers || can || mount || an || NFS || file || system || in || Ultrix || or || OSF, || even || if || it || is || denied || on || the || access || list.
Denial || of || service || in || syslog || by || sending || it || a || large || number || of || superfluous || messages.
FormMail || CGI || program || allows || remote || execution || of || commands.
The || Webgais || program || allows || a || remote || user || to || execute || arbitrary || commands.</td></tr>
</table>

Fig. 7 Pre-defined threats

mean. Subsequently, mean clusters are computed for individual clusters until the criteria are met. Based on the criteria, the predefined threats are allocated along with the consideration of information matching on the clustered data. The overall matching process is shown in Fig. 5 where pattern matching is done in blocks for the considered text.

Subsequently, the information is matched with the clustered data using the M-KMP algorithm. Then, information occurrence is assessed on the allocated threat data. If the information occurrence is found, then, the threat is marked which is updated in the predefined threat. On contrary, when the information seems to not occur on the allocated threat data, then, in this case, the iteration continues.

4 Results and discussion

The outcomes procured from the execution of the proposed work are discussed in this section along with environmental configuration, dataset description and performance metrics.

4.1 Environmental configuration and dataset description

Environmental configuration The proposed system is run on a pyspark, especially in spider IDE. It is a comprehensive and simple platform. Its overall description is provided in Fig. 6.

Dataset description In this research, Common Vulnerabilities and Exposures dataset is used along with tweets from the Twitter dataset. In this case, as the intrusion detection dataset has both attack and non-attack data, the present study considers attack data from the Common Vulnerabilities and Exposures dataset and non-attack data from the Twitter dataset to train the model effectively to classify attack and non-attack data. Thus, the threats are allocated in a pre-defined format as shown in Fig. 7.

4.2 Performance metrics

The present study considers standard metrics to evaluate the performance of the proposed work which is discussed in this section.

A. Accuracy

It denotes the computation of correct classification which is given by Eq. 1.

$$\text{Accuracy(A)} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FP} + \text{FN}} \tag{1}$$

Here, the symbols are represented as

TN	True Negative
FN	False Negative
FP	False Positive
TP	True Positive

B. Recall

It denotes the proportion of the retrieved text as well as relevant text to the proportion of relevant text and is given by Eq. 2.

$$\text{Recall(R)} = \frac{\text{RT} \cap \text{ReT}}{\text{ReT}} \tag{2}$$

Here,

RT	Relevant Text
ReT	Retrieved Text

Table 2 Results based on threshold values

Threshold values	Accuracy	Precision	Recall
0.3	94.84	94	95.53
0.35	94.99	94.41	95.97
0.4	95.69	95.01	96.01
0.45	95.41	95.22	96.25
0.5	96.78	96	97
0.55	96.61	96	96
0.6	93.74	92.31	95.14
0.65	93.14	91.45	94.75
0.7	91.67	90.97	93.87
0.75	90.98	89.52	92.57
0.8	89.56	89.12	91.65
0.85	88.47	88.9	90.94

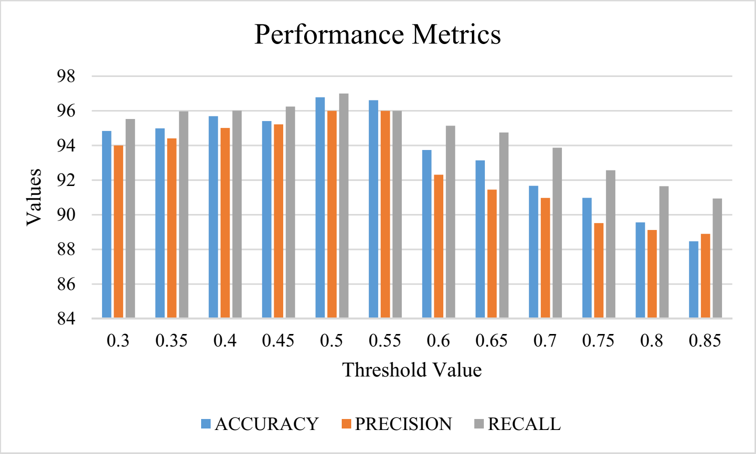


Fig. 8 Experimental outcomes based on threshold values

Table 3 Confusion matrix prediction

Confusion Matrix-prediction	Attack (–)	Non-attack (+)
Attack (–)	True Negative	False Negative
Non-attack (+)	False Positive	True Positive

Table 4 Confusion matrix based on the percentage split data

Confusion matrix for percentage split data	Fuzzy k-mean with Modified KMP	
	Attack (–)	Non-attack (+)
Attack (–)	472	15
Non-attack (+)	29	853

C. Precision

It indicates the computation of overall correct classification and is managed by incorrect classification as given by Eq. 3.

$$\text{Precision(P)} = \frac{\text{TP}}{\text{FP} + \text{TP}} \tag{3}$$

Here,

FP	False Positive
TP	True Positive

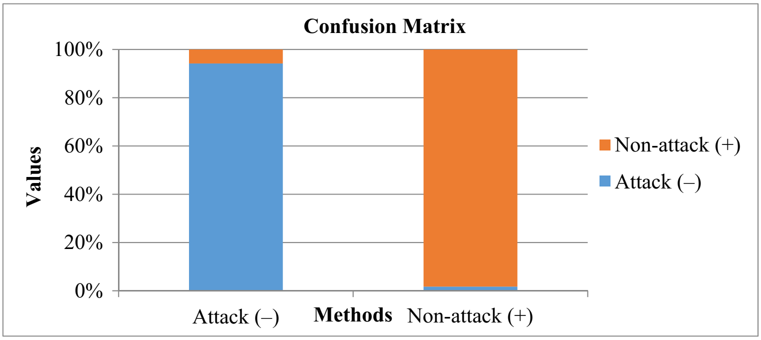


Fig. 9 Confusion matrix

Fig. 10 Receiver operating characteristic

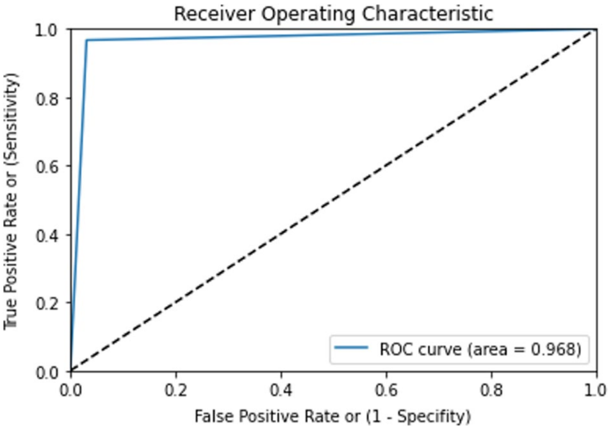


Table 5 Performance analysis

	Fuzzy k-mean with Modified KMP		
	Precision	Recall	F1 Score
Attack (–)	0.94	0.97	0.96
Non-attack (+)	0.98	0.97	0.97
Average	0.96	0.97	0.965

D. F1-score

F1-score is also called an F-measure and could be stated as the harmonic mean of Recall (R) and Precision (P). It is computed by Eq. 4,

$$\mathrm{F1-score} = \frac{2*(\mathrm{R}*\mathrm{P})}{\mathrm{R}+\mathrm{P}} \tag{4}$$

Here,

R Recall

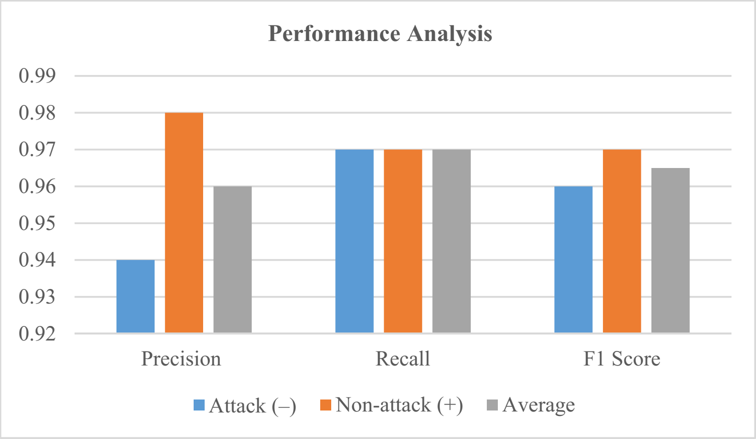


Fig. 11 Performance analysis

P Precision

Thus, by accounting these metrics for validating the model performance, these are some of the potential metrics considered.

4.3 Experimental results

The results of the proposed system based on different threshold values are given in Table 2. To explore the results, its equivalent graphical representation is shown in Fig. 8. In this case, accuracy, recall and precision are considered performance metrics.

From the results, it is found that the considered performance metrics have shown varying performance based on threshold values ranging from 0.3 to 0.85. The accuracy value is found to be high when the threshold value is 0.5 and 0.55 that is, the accuracy value is found to be 0.9678. Whereas, the precision value is found to be 0.96 and the recall value is 0.97. In addition, a confusion matrix is constructed which is used for defining the classification performance based on True Negative (TN), False Negative (FN), False Positive (FP) and True Positive (TP) as shown in Table 3.

The confusion matrix for the proposed algorithm is computed based on the percentage split data for correct and incorrect predictions of attack and non-attack. The attained analytical outcomes are shown in Table 4 with its graphical representation in Fig. 9.

From the confusion matrix, it could be found that 472 attacks are correctly classified as attacks, while 15 attacks are misinterpreted as non-attack. On contrary, 29 non-attacks are misinterpreted as attacks, whereas, 853 non-attacks are correctly classified as non-attacks. In this case, the misinterpretation rate is less than the correct classification which confirms the effective performance of the proposed system. Moreover, the ROC (Receiver Operating Characteristic) of the proposed system is shown in Fig. 10.

From the outcomes, the ROC value of the proposed system is found to be at a high rate of 0.968. In addition, the performance of the proposed system has been assessed for precision, F1-score and recall for identifying attack and non-attack. The average performance metric values are computed for this and the corresponding results are shown in Table 5 with its corresponding graph in the Fig. 11,

From the evaluation results, it is revealed that the proposed system has shown 0.96 as precision, 0.97 as recall and 0.965 as F1-score for attack and non-attack detection. These effective outcomes have been due to the innate merits of the proposed methods. Accordingly, Fuzzy K-Means permits gradual data point membership for clusters computed as degrees. This affords flexibility for exploring that, the data points could pertain to several clusters. In addition, M-KMP has the merits of confirming worst-case efficacy. The conventional Naïve search algorithm does not seem to work better when the matching characters are succeeded by any mismatching character. However, the matching seems to be better while using M-KMP. Due to these advantages, the proposed system has shown optimal outcomes in classifying intrusion detection which is confirmed through the procured outcomes.

5 Conclusion

The study aimed to accomplish intrusion detection using ML-based algorithms in a pyspark environment. Fuzzy K-means is proposed for clustering the data wherein the proposed algorithm explores data points that potentially relate to numerous clusters. It also proposed M-KMP (Modified-Knuth Morris Pratt) for accomplishing information matching on the clustered data. Based on this, the information occurrence on the allocated threat data is assessed. This process will assist in attack prevention. Experiments were performed with different threshold values ranging from 0.3 to 0.85. From the results, it was found that optimal outcomes were attained when the threshold value was 0.5 and 0.55. Analytical outcomes explored that, the proposed system explored 0.96 as precision, 0.965 as F1-score, 0.97 as recall and 0.9678 as accuracy. A confusion matrix and ROC curve were presented for the proposed system which revealed that the proposed system was capable of classifying the data correctly in comparison with the misinterpretation rate. Moreover, the ROC value was exposed to be 0.968 which confirms its effectiveness. Due to such effective outcomes, the present work will serve as a pathway for security developers to detect attacks. However, to enhance the accuracy rate, different ML-based algorithms could be employed in the future to procure better results.

Funding There is no funding for this study.

Declarations

Conflict of interest There is no conflict of Interest.

References

1. Soe YN, Feng Y, Santosa PI, Hartanto R, Sakurai K (2019) Rule generation for signature based detection systems of cyber attacks in iot environments. Bull Netw Comput Syst Softw 8(2):93–97
2. Ali MM, El-Henawy IM, Salah A (2021) Usages of spark framework with different machine learning algorithms. Comput Intell Neurosci 2021. https://doi.org/10.1155/2021/1896953

3. Othman SM, Ba-Alwi FM, Alsohybe NT, Al-Hashida AY (2018) Intrusion detection model using machine learning algorithm on Big Data environment. J Big Data 5(1):1–12
4. Morfino V, Rampone S (2020) Towards near-real-time intrusion detection for IoT devices using supervised learning and apache spark. Electronics 9(3):444
5. Singh J, Singh J (2021) A survey on machine learning-based malware detection in executable files. J Syst Architect 112:101861
6. Karataş F, Korkmaz SA (2018) Big Data: controlling fraud by using machine learning libraries on spark. Int J Appl Math Electron Computers 6(1):1–5
7. Peng K, Leung VC, Huang Q (2018) Clustering approach based on mini batch kmeans for intrusion detection system over big data. IEEE Access 6:11897–11906
8. Sun L, Zhang H, Fang C (2021) Data security governance in the era of big data: status, challenges, and prospects. Data Sci Manage 2:41–44
9. Latah M (2020) Detection of malicious social bots: a survey and a refined taxonomy. Expert Syst Appl 151:113383
10. Zhang F, Kodituwakku HADE, Hines JW, Coble J (2019) Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data. IEEE Trans Industr Inf 15(7):4362–4369
11. Do Xuan C, Nguyen HD, Tisenko VN (2020) Malicious URL detection based on machine learning. Int J Adv Comput Sci Appl 11(1). https://doi.org/10.14569/ijacsa.2020.0110119
12. Shi Y, Chen G, Li J (2018) Malicious domain name detection based on extreme machine learning. Neural Process Lett 48(3):1347–1357
13. Liu S, Huang S, Xu X, Lloret J, Muhammad K (2023) Efficient visual tracking based on fuzzy inference for intelligent transportation systems. IEEE Trans Intell Trans Syst. https://doi.org/10.1109/TITS.2022.3232242
14. Liu S et al (2022) Human inertial thinking strategy: a novel fuzzy reasoning mechanism for IoT-assisted visual monitoring. IEEE Internet of Things J 10(5):3735–3748
15. Jemal I, Cheikhrouhou O, Hamam H, Mahfoudhi A (2020) Sql injection attack detection and prevention techniques using machine learning. Int J Appl Eng Res 15(6):569–580
16. Dhalaria M, Gandotra E (2021) A hybrid approach for android malware detection and family classification. IJIMAI 6.6(2021):174–188
17. Singh J, Singh J (2020) Detection of malicious software by analyzing the behavioral artifacts using machine learning algorithms. Inf Softw Technol 121:106273
18. Shahriar H, Nimmagadda S (2021) Network Intrusion Detection for TCP/IP packets with machine learning techniques. In: Machine Intelligence and Big Data Analytics for Cybersecurity Applications. Springer, vol 919, pp 231–247. https://doi.org/10.1007/978-3-030-57024-8_10
19. Subroto A, Apriyana A (2019) Cyber risk prediction through social media big data analytics and statistical machine learning. J Big Data 6(1):1–19
20. Kotenko I, Saenko I, Branitskiy A (2018) Framework for mobile internet of things security monitoring based on big data processing and machine learning. IEEE Access 6:72714–72723
21. Rashid M, Singh H, Goyal V, Parah SA, Wani AR (2021) Big data based hybrid machine learning model for improving performance of medical internet of things data in healthcare systems. In: Healthcare Paradigms in the Internet of Things Ecosystem. Elsevier, pp 47–62
22. Shrestha R, Omidkar A, Roudi SA, Abbas R, Kim S (2021) Machine-learning-enabled intrusion detection system for cellular connected UAV networks. Electronics 10(13):1549
23. Peng K, Leung V, Zheng L, Wang S, Huang C, Lin T (2018) Intrusion detection system based on decision tree over big data in fog environment. Wirel Commun Mob Comput 2018. https://doi.org/10.1155/2018/4680867
24. Deepa G, Thilagam PS, Khan FA, Praseed A, Pais AR, Palsetia N (2018) Black-box detection of XQuery injection and parameter tampering vulnerabilities in web applications. Int J Inf Secur 17(1):105–120
25. Atefinia R, Ahmadi M (2022) Performance evaluation of Apache Spark MLlib algorithms on an intrusion detection dataset. J Comput Secur 9(1):57–69
26. Marir N, Wang H, Feng G, Li B, Jia M (2018) Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark. IEEE Access 6:59657–59671
27. Hafsa M, Jemili F (2018) Comparative study between big data analysis techniques in intrusion detection. Big Data and Cogn Comput 3(1):1
28. Donkal G, Verma GK (2018) A multimodal fusion based framework to reinforce IDS for securing Big Data environment using spark. J Inform Secur Appl 43:1–11
29. Atefinia R, Ahmadi M (2021) Network intrusion detection using multi-architectural modular deep neural network. J Supercomput 77(4):3571–3593

30. Basnet RB, Shash R, Johnson C, Walgren L, Doleck T (2019) Towards detecting and classifying Network Intrusion Traffic using Deep Learning frameworks. J Internet Serv Inf Secur 9(4):1–17
31. Al-Tarawneh A, Al-Saraireh Ja (2021) Efficient detection of hacker community based on twitter data using complex networks and machine learning algorithm. J Intell Fuzzy Syst 40(6):12321–12337
32. Islam U et al (2022) Detection of Distributed Denial of Service (DDoS) Attacks in IOT Based Monitoring System of Banking Sector Using Machine Learning Models. Sustainability 14(14):8374
33. Iqbal F, Batool R, Fung BC, Aleem S, Abbasi A, Javed AR (2021) Toward tweet-mining framework for extracting terrorist attack-related information and reporting. IEEE Access 9:115535–115547
34. Bouya-Moko BE, Boahen EK, Wang C (2022) Fuzzy Local Information and Bhattacharya-Based C-Means Clustering and Optimized Deep Learning in Spark Framework for Intrusion Detection. Electronics 11(11):1675
35. Gupta R, Tanwar S, Tyagi S, Kumar N (2020) Machine learning models for secure data analytics: A taxonomy and threat model. Comput Commun 153:406–440
36. Akkem Y, Biswas SK, Varanasi A (2023) Smart farming using artificial intelligence: a review. Eng Appl Artif Intell 120:105899

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Gousiya Begum[1,2] · S. Zahoor Ul Huq[3] · A. P. Siva Kumar[4]

S. Zahoor Ul Huq
szahoor@gmail.com

A. P. Siva Kumar
sivakumar.ap@gmail.com

[1] Research Scholar, Department of Computer Science and Engineering, JNTUA Anantapur, Anantapuramu, Andhra Pradesh, India

[2] Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Gandipet, Hyderabad, India

[3] Department of Computer Science and Engineering, G. Pulla Reddy Engineering College, Kurnool, Andhra Pradesh, India

[4] Department of Computer Science and Engineering, JNTU Anantapur, Anantapuramu, Andhra Pradesh, India