International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August 2019

Detection of Intrusion using Hybrid Feature Selection and Flexible Rule Based Machine Learning

B. Sudhakar, V. B. Narsimha, G. Narsimaha

Abstract: ***With the rapid growth in the data processing and data sharing, the application owners and the consumers of the applications are more influenced to use the remote storage on cloud-based data centre and the application generated data is also growing ups and bounds. Nevertheless, the adaptation of the data sharing, and data processing applications were not easy for the consumers. The application owners and the service providers have struggled with the sensitive data of the consumers and the consumers were also faced trust issues with the complete framework. The standard legacy applications were designed for the traditional centralized scenarios, where the intrusion detection can be performed only using the network status analysis and the application characteristics analysis. Moreover, most of the parallel calculations initially enhance the hybrid likelihood and change likelihood of GA as indicated by the populace advancement variable-based math and wellness esteem. Nevertheless, the population of data and the attacks on the data is high and the correct population size is highly difficult to determine. Regardless to mention, that the use of fitness functions will restrict the attack detection to certain types and these algorithms are bound to fail in case of a newer attack. However, with the migration of application to the data processing framework, the consumers have started demanding more security against the intrusions. A good number of research attempts were made to map the traditional security algorithms into the data processing space, nonetheless, the attempts were highly criticized due to the lack of proper analysis of security attacks on data processing applications. Hence, this work proposes a novel framework to detect the intrusions on data processing framework with justifying attack characteristics. This work proposes a novel algorithm to reduce the features of attack characteristics to justify the gaps on data processing frameworks with significant reduction in time for processing and further, proposes an algorithm to derive a strong rule engine to analyse the attack characteristics for detecting newer attacks. The complete proposed framework demonstrates nearly 93% and higher accuracy, which is much higher than the existing parallel research outcomes with least time complexity.***

Keywords : Attack Characteristics, Dynamic Rule engine, Hybrid Feature Selection, IDS, Knowledge-based feature

Revised Manuscript Received on August 30, 2019.

* Correspondence Author

B. Sudhakar *, Reserach Scholar-JNTUH & Associate Professor, Department of CSE-GNIT

V. B. Narsimha, Associate Professor, Department of CSE-University College of Engineering, OU

Dr. G. Narsimaha, Professor, Department of CSE- JNTUH College of Engineering, JNTUH.

I. INTRODUCTION

With the advancement and promotion of data and network advances, network data security is winding up increasingly essential. Contrasted and customary network defence technology, (for example, firewalls), human-focused shrewd IDSs that can step up with regards to catch and caution of network intrusion has an incredible useful esteem. The topic of how to enhance the viability of brilliant network intrusion location has turned into a focal point of network security. The recommendations from the notable survey work by A. Sultana et al. [1] have highlighted the benefits of traditional machine learning methods for intrusion detection for network centric or centralized applications. Notwithstanding to mention that these algorithms have failed to address the data processing security challenges. At present, the utilization of brilliant IDS is seen as a compelling answer for network security and insurance against outside dangers. Be that as it may, the current IDS regularly have a lower identification rate under new assaults and has a high overhead when working with review information, and in this manner machine learning strategies have been generally connected in interruption location. SVM, one of the machine learning advances, is another calculation dependent on statistical learning theory that has demonstrated higher execution than the conventional learning techniques in taking care of the classification issue of social data processing problems. The notable work by L. Oneto et al. [2] defining the limits of data classifications for social media data. A good number of research attempts were made to apply the similar mechanisms for intrusion attack detection and the works were criticized for the solutions being attack centric and not able to comply with the newer attacks.

Contrasted and other grouping calculations, SVM can more readily tackle the issues of little examples, nonlinearity, and high dimensionality. Be that as it may, with the approach of the period of enormous information, SVM experiences the issue of long preparing and testing times, high blunder rates and low obvious positive rates, which constrain the utilization of SVM in system interruption identification. Along these lines, SVM highlight determination, include weighting, and SVM parameter setting are basic to enhanced identification execution. Henceforth, the demand of a unique algorithm for detecting the intrusion attacks are becoming the demand of the modern research. Thus, this paper proposes a novel flexible rule-based attack characteristics detection method with a novel feature reduction method.

Henceforth, the rest of the work is organized as in the Section – II, the outcomes of the parallel researches are analysed, in Section – III the data processing attack characteristics were analysed, in Section – IV, the problem is formulated and the expected outcomes were analysed, in Section – V the first outcome of this work as feature reduction is proposed, in the Section – VI the second outcome as flexible rule based intrusion detection algorithm is proposed, further in Section – VII the results obtained from this framework is analysed, the comparative analysis is presented in Section – VIII and in the Section – IX the work presents the final conclusion.

II. OUTCOME OF THE PARALLEL RESEARCH WORKS

In the period of enormous information, interruption discovery has turned into the most imperative theme in security framework. To recognize assault and ordinary system get to, various machine learning strategies are connected in IDS, including fluffy rationale. The work of A. Chaudhary et al. [3] have successfully demonstrated the implementation of the fuzzy logic-based intrusion detection system for traditional scenarios. Also, the genetic approaches for IDS are very much successful as reported by S. Malhotra et al. [4]. Nonetheless, the works by R. Sen et al. [5] using artificial neural network, M. Tabatabaefar et al. [6] using artificial immune systems and the work by T. Mehmood et al. [7] using ant colony optimization have demonstrated higher accuracy but at the cost of higher time complexity and at the compromise of newer attack detection. Elaborating the same, although SVM based IDS can enhance IDS execution regarding identification rate and learning speed contrasted and conventional calculations, opportunity to get better still exists. As the quantity of highlights of the review information ends up bigger, the execution of IDS debases regarding preparing time and grouping precision.

The work of M. S. Pervez et al. [8] shows the use of the hereditary calculation was proposed to enhance the interruption location framework in view of support vector machine, and the ideal element subset was chosen for SVM. Be that as it may, the blunder rate of SVM was not considered. The proposed algorithm by Y. Guang et al. [9] is an intrusion detection technique dependent on wavelet part minimum square was intended to enhance the discovery ability of SVM in complex nonlinear frameworks. In any case, the preparation and testing time of the calculation is generally long. In the work of T. Yerong [10], the heuristic hereditary calculation was connected to upgrade the SVM portion parameters. The hereditary administrator is powerfully balanced through a heuristic system, and the characterization precision of the model is taken as the target capacity to acknowledge parameter advancement of the Gaussian part based SVM arrangement display. In any case, this methodology did not consider the effect of highlight weighting on SVM recognition exactness. Analysing the work of Z. Chen et al. [10], it is natural to realize that the coarse-grained parallel hereditary calculation was exhibited to at the same time streamline the component subsets and parameters of SVM. Another wellness work was recommended that incorporates the characterization exactness, the quantity of highlights and the quantity of help vectors, however it required quite a while to prepare the SVM.

In the other hand, the work by K. S. Desale et al. [12] shows the use of GA, which is chosen as a standout amongst the most ground-breaking instruments to look in a huge space with the possibility to locate the best arrangement in the pursuit space. Notwithstanding, in the later advancement of the populace, a bigger hybrid and change likelihood may result in the loss of good qualities and postponed intermingling of the calculation.

In synopsis, albeit numerous SVM-based system intrusion location strategies have been proposed as of late, the above calculations still experience the ill effects of specific inadequacies:

- **Redundancy of the Dataset**: Because of excess features, the crude dataset befuddles the classifier, prompting off base discovery. Conventional element choice overlooks various delicate highlights, bringing about a classifier without ideal affectability.
- **Lack of Applicability**: The traditional IDSs cannot match the higher complexity of the attacks causing higher losses to the applications running on data processing framework.
- **Higher Time Complexity:** On the off chance that GA is utilized to advance the SVM-based intrusion recognition framework, the preparation time is longer, and the mistake rate is higher while choosing the ideal element subset. After choosing the ideal component subset, the significance of the highlights isn't arranged.

Hence, this work proposes a novel algorithm as the first component to reduce or select appropriate features and further, proposes another algorithm component to build a rule-based engine to detect the newer types attacks.

However, firstly the intrusion attack characteristics for data processing framework must be analysed in order to identify the accurate features from the dataset based on the identified characteristics. Thus, the next section of this work, elaborates on the attach characteristics on data processing framework and the data processing framework as well.

III.ATTACK CHARACTERISTICS FOR DATA PROCESSING FRAMEWORK

The intrusion attacks on the traditional systems are different than the attacks on the data processing architecture. Thus, in order to develop an ideal intrusion detection system for data processing frameworks demands a detailed study of the data processing framework in general and study of the attack types on data processing. Henceforth, in this section of the work, the data processing architecture and the attack characteristics are analysed.

In the Data processing time, calculation and capacity are shoddy per TB. Along these lines, with consistently developing computational capacities, framework use is no longer as basic a factor. It is currently plausible to utilize more computational capacity to do a similar work [Fig – 1]. In the meantime, the measure of data that needs handling has been expanding exponentially in the previous decade because of upgrades in data age and capacity limit [13]. Most importantly, programming apparatuses and procedures have developed with globalization and the Internet. It is progressively attainable to reuse code, thusly, the centre has moved to incorporate codes made by various networks.

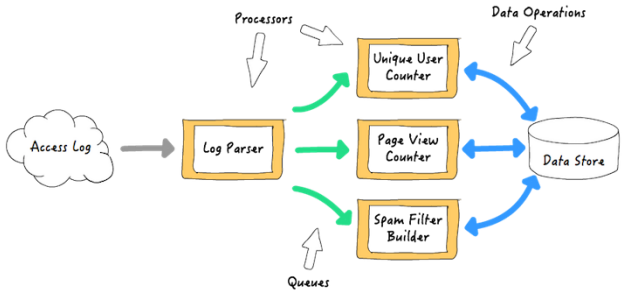

Fig. 1 Data processing Framework [14]

Elite system limit, that gives the spine to the top of the line registering frameworks, has not expanded at an indistinguishable rate from preparing and capacity abilities. Subsequently, the limitation in calculation has just moved from moving data to a big supercomputer, to moving the application to numerous littler PCs where the data lives. Programming such a methodology isn't new, the application is executed where the data is kept in an inexactly coupled and very dispersed design as portrayed by N. Chand et al. [15]. Conversely, Relational Database Management Systems (RDBMS) will, in general, give access to data as one Data processing storehouse dependent on effective firmly coupled frameworks. Organized Query Language is the accepted technique to get to databases as it gives moderately simple access to data at various dimensions inside associations. Usually to see low-level software engineers and abnormal state business examiners having a similar bit of SQL and comprehension or endeavouring to comprehend it. This sharing model has its impediments and can't misuse furthermore, handle the monstrous increment in static non-evolving data.

As of late, there has been an expansion in NoSQL ways to deal with defeat these shortcomings as distinguished by R. Vijayanand et al. [16]. Regardless of their generally late rise, there are currently more than one hundred NoSQL approaches that have practical experience in administration of various multimodal data types (from organized to non-organized) and with the expected to understand unmistakable difficulties. Most are controlled by the Map-Reduce worldview that originated from Google, which depends on a greatly appropriated design that misuses modest ware equipment. Therefore, the requirement for the production component for putting away and preparing data is dispensed with. It is in truth less expensive to copy and to over computer as correspondence is moderately more costly than capacity and computational assets.

Further in this section of the work, the intrusion attack characteristics are analysed.

A. **Infrastructure Security**: In the data processing domain, the submitted jobs are processed in distributed architecture. The submitted jobs are segregated into two major parts as mappers and reducers. The reducer jobs are distributed to the slave nodes and further be processed. Here, the faulty slave nodes can introduce intrusions. The fundamental property to identify the intrusion from the infrastructure security can easily be identified based on the number of connection requests from slave nodes to the master node as it exceeds the number of reducers.

This ideology can be inferred as:

$$J = \sum_{i=1}^{n} M_i + \sum_{i=1}^{n} R_i \qquad \text{(Eq. 1)}$$

Here, considering any submitted job (J) can maximum be broken into collection of two parts as mappers and reducers. Here the reducers must be deployed into the slave nodes in order to complete.

Further, the number of reducer job (η)

$$\phi(R_i) \rightarrow \eta \qquad \text{(Eq. 2)}$$

Also, during the execution of the reducer jobs, the slave nodes will make number of connection requests (λ) to the mapper on master node.

$$\prod_{conn=Serv\_Req} \rightarrow \lambda \qquad \text{(Eq. 3)}$$

It is natural to realize that the number of connection requests (λ) are to be equal to the number of reducer job (η) as:

$$\lambda \approx \eta \qquad \text{(Eq. 4)}$$

Hence, in order to identify the intrusion attacks from infrastructure view point, the following properties are to be taken care [Table – 1].

TABLE I: INFRASTRUCTURE SPECIFIC ATTACK CHARACTERISTICS

SNO	Characteristics Type	Description
1	Count_Connection_req	Number of Connection requests from the Slave nodes to the master node
2	Count_Reducer	Number of reducer jobs at slaves (Mean Value)
3	Ratio	Count_Connection_req / Count_Reducer

B. **Data Privacy**: During the data processing operations, the privacy of the data plays a major role. The exposure of the information to slave nodes can be of higher challenge to comply. The slave nodes are often under the control of third party users and protecting data privacy can be highly difficult. The intrusion attacks can be identified based on the protocol types as for a specific type of application data, specific type of network protocols can be adopted and any violation to the network protocol types can lead to the changes of introducing attacks. Also, the validation of the client logins at the slave nodes or the reducer nodes must be authenticated in order to reduce the changes of attacks.

Further to justify the characteristics, the valid network protocol (NVP) must comply as,

$$NVP = \begin{cases} If, \prod_{Data\_Privacy=True,UniCast} = TCP \\ Else, \prod_{Data\_Privacy=False,BroadCast} = UDP \end{cases} \qquad \text{(Eq. 5)}$$

Hence, in order to identify the intrusion attacks from data privacy view point, the following properties are to be taken care [Table – 2].

TABLE II: DATA PRIVACY SPECIFIC ATTACK CHARACTERISTICS

SNO	Characteristics Type	Description
4	Connection_Type	The network protocol during the communication from reducer node to mapper
5	Access_type	Identification of guess access if any.
		from the reducer nodes to the mapper or the master node
6	Rate_of_Serv_Type_Change	Rate of the change in service request types from the reducer nodes to the mapper or the master node

C. **Data Integrity Management**: The applications submitted to the network for data processing handles huge amount of data, which must be read, processed and further updated. During the process of update, it is often being observed that the integrity of the data is compromised, which can be intentional by intrusion attacks. The attacks can be identified by analysing the service request types from the reducer or slave nodes. It is natural to realize that the specific service types are forbidden for specific data or information during the update operations and can cause the integrity to lose. Any well configured client running on the reducer nodes must not run any service which may cause integrity issues on the data. Reversibility, presence of incorrect service types can be identified as intrusion attacks.

Thus, the integrity protector (IP) can be defined as,

$$IP = \begin{cases} If\ (Data = \mathrm{Re}\,adOnly) \rightarrow HTTP \\ Else, \rightarrow Te\ln et \end{cases} \quad \text{(Eq. 6)}$$

Hence, in order to identify the intrusion attacks from data integrity view point, the following properties are to be taken care [Table – 3].

TABLE III: DATA INTEGRITY SPECIFIC ATTACK CHARACTERISTICS

SNO	Characteristics Type	Description
6	Service_Request_type	The request types

D. **Reactive Security**: During the application and data processing in any data processing framework, the network characteristics must be identified with close observations. Many of the times it is being observed that, the intrusion attacks include higher amount of network traffic to reduce the performance capabilities of the complete framework. The generic nature of these attacks is to include higher priority packets in the network requests. Thus, identifying the flow rate of the higher

E. priority network packets may lead to reduce the changes of intrusion attacks and provide reactive security.

Hence, in order to identify the intrusion attacks from reactive security view point, the following properties are to be taken care [Table – 4].

TABLE IV: REACTIVE SECURITY SPECIFIC ATTACK CHARACTERISTICS

SNO	Characteristics Type	Description
7	Rate_of_priority_packets	The amount of priority data packets is introduced in the network

Henceforth, after identifying the attack characteristics, this work provides the mapping of few popular attack types with these characteristics [Table – 5].

TABLE V: PROPOSED ATTACK TYPES AND CHARACTERISTICS MAPPING

Attack Type	Attack Identifier Characteristics						
	Count_Connection_req	Count_Reducer	Ratio	Connection_Type	Access_type	Service_Request_type	Rate_of_Serv_Type_Change
Browser attacks	Yes				Yes		
Brute force attacks	Yes	Yes	Yes				
Denial of service attacks					Yes	Yes	Yes
SSL attacks						Yes	Yes
Scans			Yes				Yes
DNS attacks						Yes	

Here it is natural to understand that apart from the specific attacks, this proposed characteristic mapping metric can also identify any newer types of attacks by identifying violation of any of the mentioned characteristics.

Further, the intrusion attack identification can only be possible by analysing the complete network behaviour. The data processing network can return a huge number of parameters and the existing datasets must be reduced to comply with the lower time complexity. Also, the intrusion detection must be done using a novel rule engine, which can infer newer rules to identity previously unidentified intrusion attacks.

Henceforth, with the identification of each attack characteristics, this work in the next section formulates the problem.

IV. PROBLEM FORMULATION

In this section of the work, the problem is formulated. The primary purpose of this research is to reduce the time complexity of the intrusion detection system and infer dynamic rules for newer intrusion detection. To establish the objectives and goal of the research, this work furnishes two lemmas in this section.

Lemma – 1: The reduced number of attributes without losing the knowledge of the dataset reduces the time complexity of the detection or prediction system.

Where,

A, the set of initial attributes as $\{a_1, a_2, a_3, a_4, ...a_n\}$ for total of n attributes

K, the set of initial knowledge from each attribute as $\{k_1, k_2, k_3, k_4, ...k_n\}$ for total of n attributes

T is the set of time complexity to consider each attribute as $\{t_1, t_2, t_3, t_4, ...t_n\}$ for total of n attributes

Proof: To prove the above lemma, the work demonstrates the followings:

Firstly, the combined knowledge from the dataset can be considered and the equivalence of knowledge can also be identified as

$$K_i = K_j \qquad \text{(Eq. 7)}$$

Where, i and j denotes two random variables or attributes in the dataset.

Secondly, the total time complexity of the dataset processing for detection or prediction can be identified as the total time complexity for including each element, T_1,

$$T_1 = \sum_{i=1}^{n} t_i \qquad \text{(Eq. 8)}$$

Further, by the principle of Eq. – 7, considering z, the number of attributes contribute similar knowledge for detection and prediction. Hence, the reduced size of the dataset A' is

$$A' = \{a_1, a_2, a_3, ...a_z\} \qquad \text{(Eq. 9)}$$

Henceforth, the total time complexity of the reduced set can be considered as T_2,

$$T_2 = \sum_{i=1}^{z} t_i \qquad \text{(Eq. 10)}$$

As, $A' < A$, hence it is natural to realize that,

$$T_2 < T_1 \qquad \text{(Eq. 11)}$$

And

$$(K' = \sum_{i=1}^{z} k_i) = (K = \sum_{i=1}^{n} k_i) \qquad \text{(Eq. 12)}$$

Henceforth, it is simple to justify that reducing the size of the dataset in terms of dimensions or number of attributes can reduce the time complexity without losing the dataset knowledge for detection or prediction.

Thus, the first problem this research addresses is to reduce the dataset dimension without any information loss.

Lemma – 2: The inferable rulesets from any rule engine is more effective to detect more number of attacks from any dataset.

Where,

A is the set of identified and known attacks can be defined as $\{a_1, a_2, a_3, a_4, ...a_n\}$

C is the set of characteristics to define and identify the attack set A as $\{\sum_{i=1}^{k_1} c_i, \sum_{i=1}^{k_2} c_i, \sum_{i=1}^{k_3} c_i, \sum_{i=1}^{k_4} c_i, ..., \sum_{i=1}^{k_n} c_i,\}$ with k denoting number of characteristics for each attack

R denotes the set of rules to detect each attack, considering each attack needs F number of rules to be present to justify c number of characteristics as $\{F_1, F_2, F_3, F_4....F_n\}$

Proof: To prove the above lemma, the work demonstrates the followings:

First each attack can be defined as,

$a_1 = \sum_{i=1}^{k_1} c_i \rightarrow c_1$ and sub sequentially the other attacks as well.

Thus, it is natural to infer that, every characteristics set must have some common characteristics as all the characteristics are part of the major characteristics set C.

This principle of understanding can be denoted as,

$$\sum_{i=1}^{k_1} c_i \subseteq \sum_{i=1}^{k_2} c_i \qquad \text{(Eq. 13)}$$

Hence, the rulesets defining the characteristics can also possibly be inferred as,

$$F_1 \subseteq F_2 \qquad \text{(Eq. 14)}$$

As, the Eq. 14 stands true for all the cases where Eq. 13 stands true, thus it is natural to understand that, the rules part of each rulesets can be further inferred from each other, as

$$F_1 = \{r_1, r_2, r_3, ..., r_n\}$$
$$F_2 = \{r_1', r_2', r_3', ..., r_n'\} \qquad \text{(Eq. 15)}$$

And

$$r_1 \rightarrow r_1' \qquad \text{(Eq. 16)}$$

Thus, the inferable rules available in the ruleset can easily identify newer characteristics and further can identify newer attacks.

Henceforth, the second problem this research addresses is to define a dynamic inferable ruleset for detecting unidentified and newer attacks with any static characteristics sets.

In the next section of this work, the first problem, this research identifies, knowledge lossless feature reduction proposed algorithm is furnished.

V. PROPOSED FEATURE SELECTION ALGORITHM

In this section of the work, the algorithm for selecting and reducing the dataset is used. Based on the formulation of the problem, the reduction of the features can significantly reduce the time complexity of the dataset analysis for detection of intrusions.

In the attack characteristic identification phase, this work has demonstrated that the dataset must contain features which are related to network infrastructure, data security, data integrity management and finally the reactive security. This algorithm analyses the dataset and extracts the features related to these knowledges. The proposed algorithm is furnished here:

Algorithm I: Knowledge Based Feature Subset Selection (KBFSS)

Step - 1. Accept the dataset
Step - 2. Define the knowledge set as network infrastructure (NI), data security (DS), data integrity management (DIM) and reactive security (RS)
Step - 3. Identify the features from the dataset and make a subgroup for NI
Step - 4. Identify the features from the dataset and make a subgroup for DS
Step - 5. Identify the features from the dataset and make a subgroup for DIM
Step - 6. Identify the features from the dataset and make a subgroup for RS
Step - 7. For each element in NI
  a. For each element in DS
    i. For each element in DIM
      1. For each element in RS
        a. Build the possible combination set.
Step - 8. For each combination find MAE
Step - 9. Find the feature subset with lowest MAE
  a. Find the equivalent to Knowledge subset and report the final feature subset

The result from the proposed algorithm is a reduced feature subset containing at least one parameter from each knowledge set and elaborated in the further section of this work.

Further, in the next section of the work, the proposed rule-based intrusion detection algorithm is proposed.

VI. PROPOSED DYNAMIC RULE BASED INTRUSION DETECTION MECHANISM

In this section of the work, the dynamic rule-based intrusion detection algorithm is discussed. The algorithm is designed to include maximum inferences possible from minimum actual rules available. Henceforth, the objective is to design an algorithm to extract minimum number of rulesets with maximum number of features included. Once the primary rulesets are built, the proposed algorithm extracts most inferable rules from the initial ruleset and builds the final minimal ruleset with maximum inferable rules. Further, the same rules can be deployed to detect the intrusion at the data processing networks.

The proposed algorithm is furnished here:

Algorithm II: Dynamic Inferable Rule Engine for Intrusion Detection (DIREID)

Step - 1. Accept the dataset and extract feature set from metadata
Step - 2. Accept the class variable feature from the metadata
Step - 3. For each ItemSet in the dataset
  a. Build confidence function, CF = ItemSet(no_of_features without null or missing values) and class_variable = TRUE
  b. if CF > 0,
    i. Then build ruleset = no_of_features with Itemset value
Step - 4. For each ruleset
  a. Calculate the number of predicates
  b. For each predicate
    i. If predicate[i] can be inferred from predicate[j] and ruleset contains predicate[i] or predicate[j]
    ii. Then count the number of inferable predicates as psum
    iii. If psum > threshold
    iv. Then accept the rule
Step - 5. Build the final ruleset with highest psum values for each rule
Step - 6. Accept the test dataset
Step - 7. For each instance
  a. If the existing ruleset can find the intrusion
    i. Then mark the connection as attack
  b. If the instance cannot be validated by existing ruleset
    i. Then infer the new rule and repeat Step – 7.a
Step - 8. Report all attacks

The outcome of this proposed algorithm is the minimum rulesets and as the newer rules can be inferred from this minimal ruleset, the newer attacks also can be detected for which no sufficient information is present.

In the next section of the work, the results are discussed.

VII. RESULTS AND DISCUSSION

The results obtained from the proposed algorithms are highly satisfactory and this section of the work discusses the results from each phase of the proposed algorithm.

A. Dataset Analysis – Applicability for Modern Intrusion Detection

Firstly, the analysis of the dataset is carried out in this part. The used training and testing dataset is KDD CUP 99 Data Set [17]. The dataset, being perfect for intrusion detection, is a popular choice for many researchers.

Nevertheless, this work initially analyses the dataset to check the applicability for intrusion detections on data processing framework.

This work analyses the availability of the features for intrusion detection in data processing framework based on the studies conducted at the Section – III of this work.

The mapping for feature category and availability is furnished here [Table – 6].

TABLE VI: FEATURE AVAILABILITY ANALYSIS ON KDD

Feature Class	Number of Features available form KDD	List of Features
Infrastructure	9	Serial No. 1 to 9
Data Security	13	Serial No. 10 to 22
Data Integrity	9	Serial No. 23 to 31
Responsive	10	Serial No. 32 to 41

The feature availability is visualized graphically here [Fig – 2].

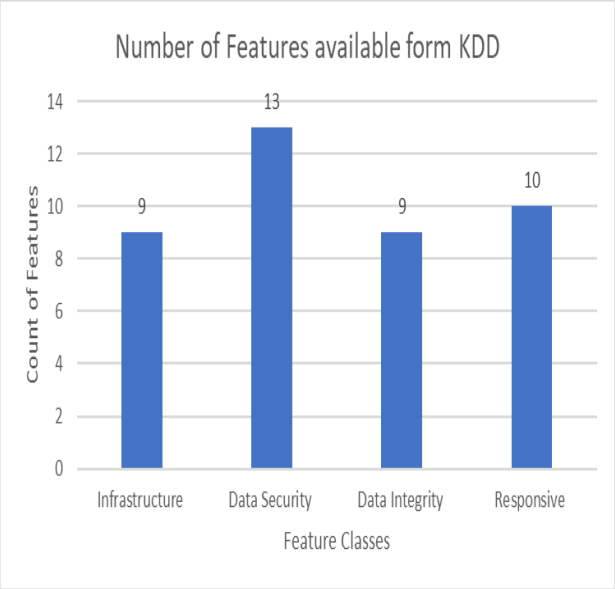

Fig. 2 Feature Availability Analysis

Hence, it is natural to understand that the KDD dataset is highly applicable for detecting intrusion in the data processing framework.

B. Feature Set Selection

Secondly, this work formulates the results from the KBFSS algorithm presented in this work. The phase wise results from the algorithm is presented here [Table - 7]:

TABLE VII: FEATURE SELECTION ALGORITHM PASS RESULTS

Pass Number	Number of Feature sets evaluated	Accuracy = 100 - (Total MAE / Total Instances) * 100	Data processing framework intrusion Detection Features	Other Features
0	41	55.95475715236194	1	1
1	40	87.18119316921712	2	2
2	49	90.1973830117542	4	3
3	38	91.70547793302285	4	4
4	37	92.37081392770015	3	4
5	36	93.08050565535595	6	0
6	35	93.2579285872699	6	1
7	34	93.2579285872699	6	2
8	33	93.2579285872699	6	3
9	32	93.2579285872699	6	4
10	31	93.2579285872699	5	6

Hence, it is natural to realize that in the pass – 5, this algorithm finds all the data processing framework relevant features with minimal irrelevant features with highest accuracy in the class.

The process result is also visualized graphically here [Fig – 3].

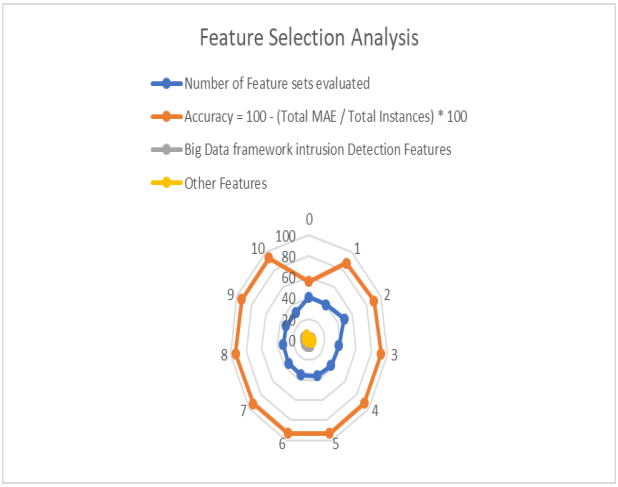

Fig. 3 Feature Selection Algorithm Analysis

Further, the final selected feature set is presented here [Table – 8].

TABLE VIII: REDCUED FEATURE AFTER SELECTION

Feature Name	Feature Class
Protocol_type	Infrastructure
Service	Infrastructure
Urgent	Infrastructure
Is_guest_login	Data Security
Count	Data Integrity
Dst_host_diff_srv_rate	Reactive Security

C. Dynamic Rule Generation

In this section of the work, the dynamic inferable rule generation algorithm is analyses. The dynamic rule generation algorithm, generates the minimal ruleset with maximum rule inferences possible [Table – 9].

TABLE IX: DYNAMIC RULE GENERATION ALGORITHM PASS

Rule Number	Rule Description	Number of Possible Inferences	Sample Inferences
1	service = ftp_data AND dst_host_diff_srv_rate <= 0.27	4	dst_host_diff_srv_rate > 0.14 AND count > 1 dst_host_diff_srv_rate <= 0.24 dst_host_diff_srv_rate > 0.19 dst_host_diff_srv_rate > 0.21
2	dst_host_diff_srv_rate > 0.14 AND count > 19 AND protocol_type = tcp AND service = private	2	dst_host_diff_srv_rate > 0.5 dst_host_diff_srv_rate > 0.58 AND count > 59
3	count <= 134 AND service = http AND dst_host_diff_srv_rate > 0	3	dst_host_diff_srv_rate <= 0.99 dst_host_diff_srv_rate <= 0.01 count > 7
4	service = ftp_data AND dst_host_diff_srv_rate > 0.14 AND count <= 2	2	count > 1 AND dst_host_diff_srv_rate <= 0.75 dst_host_diff_srv_rate > 0.28
5	service = private AND protocol_type = udp AND count <= 5	2	count > 1 AND dst_host_diff_srv_rate > 0 dst_host_diff_srv_rate <= 0.01
6	service = private AND protocol_type = udp AND count <= 5	2	dst_host_diff_srv_rate > 0 count > 1 AND dst_host_diff_srv_rate <= 0.01
7	dst_host_diff_srv_rate <= 0.01 AND count <= 149	2	service = http AND count > 2 count > 7
8	service = private AND protocol_type = udp AND count <= 5	1	dst_host_diff_srv_rate > 0 AND count <= 1
9	service = private AND protocol_type = udp AND count <= 5 AND dst_host_diff_srv_rate > 0	1	dst_host_diff_srv_rate <= 0.01

Henceforth, it is simple to realize that the number of rules in the ruleset are 9 and possible number of inferred rules are 19. Thus, the rule generation framework can build a total number of rules as 28 in the actual ruleset.

D. Intrusion Detection Analysis

This section of the algorithm analyses the intrusion detection accuracy by the proposed algorithms on the KDD dataset and the accuracy analysis is carried out here [Table – 10].

TABLE X: INTRUSION DETECTION

Analysis Parameter	Value	Percentage
Correctly Classified Instances	21022	93.2488 %
Incorrectly Classified Instances	1522	6.7512 %
Kappa statistic	0.8619	-
Mean absolute error	0.1044	-
Root mean squared error	0.2364	-
Relative absolute error	-	21.2929 %
Root relative squared error	-	47.7471 %
Total Number of Instances	22544	-

The accuracy of the proposed detection algorithm is also analysed visually here [Fig – 4].

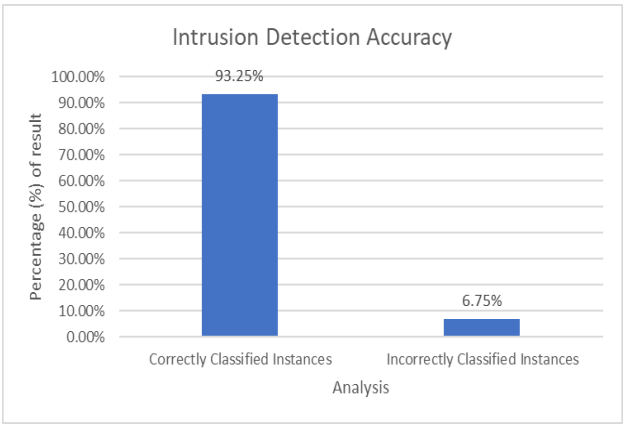

Fig. 4 Accuracy Analysis

Further, the detailed accuracy by detection classes are identified [Table – 11].

The accuracy analysis is also analysed visually here [Fig – 5].

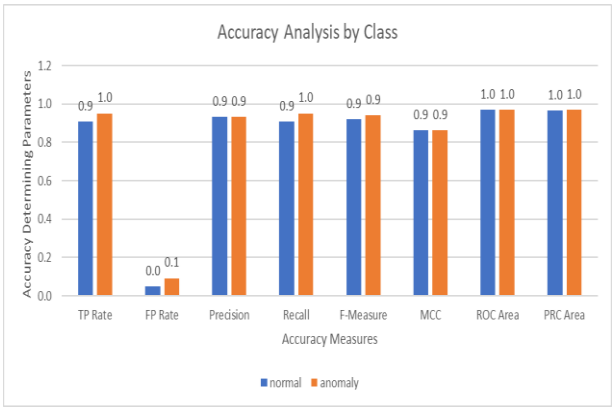

Fig. 5 Accuracy Analysis by Class

.

TABLE XI: ACCURACY ANALYSIS BY INTRUSION DETECTION CLASS

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
normal	0.908	0.049	0.934	0.908	0.921	0.862	0.971	0.964
anomaly	0.951	0.092	0.932	0.951	0.941	0.862	0.971	0.971
Weighted Avg.	0.932	0.073	0.933	0.932	0.932	0.862	0.971	0.968

Furthermore, in the next section of the work, the comparative analysis is carried out.

VIII. COMPARATIVE ANALYSIS

The comparative analysis is one of the most important section of any research in order to establish the thought that the proposed algorithms are better than the other existing algorithms or models or frameworks. Firstly, the comparative analysis is not limited to the comparing the accuracy of the algorithms, rather it is often being observed that most of the highly accurate algorithms are also highly time complex. Secondly, the many of the algorithms shares similar accuracy but one of those algorithms outperforms the other algorithms due to lesser time complexity. Henceforth, firstly the accuracy of some of the popular feature selection methods are compared here [Table – 12]. The proposed method for feature selection and few of the most popular feature reduction and selection algorithms are applied on the KDD dataset and the accuracy of detecting intrusions are tested.

TABLE XII: COMPARATIVE ANALYSIS – ACCURACY

Method	Accuracy (%)	Number of Attribute
Sequential backward floating selection	86.78	24
Sequential backward selection	86.78	17
Sequential floating forward selection	92.94	6
Proposed KBFSS algorithm	92.94	6

The results are also analysed visually [Fig – 6].

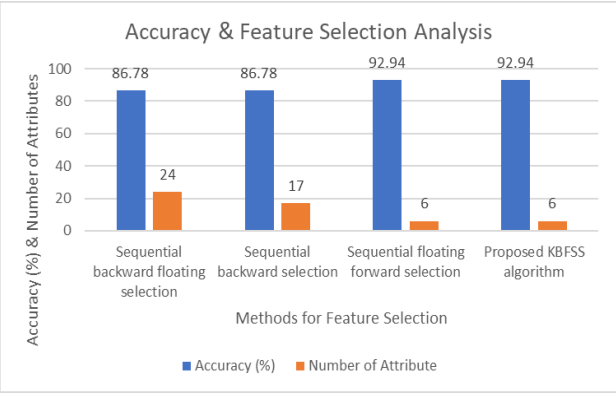

Fig. 6 Accuracy & Feature Selection Analysis

Hence, from this view point of the analysis, the proposed KBFSS algorithm and the traditional Sequential floating forward selection may provide similar accuracy. Nevertheless, the time complexity analysis must be carried out in order to realize the improvements [Table – 13].

TABLE XIII: COMPARATIVE ANALYSIS – TIME

Method	Accuracy (%)	Time Complexity (Nano Sec)
Sequential backward floating selection	86.78	88
Sequential backward selection	86.78	52
Sequential floating forward selection	92.94	13
Proposed KBFSS algorithm	92.94	11

The results are also analysed visually [Fig – 7].

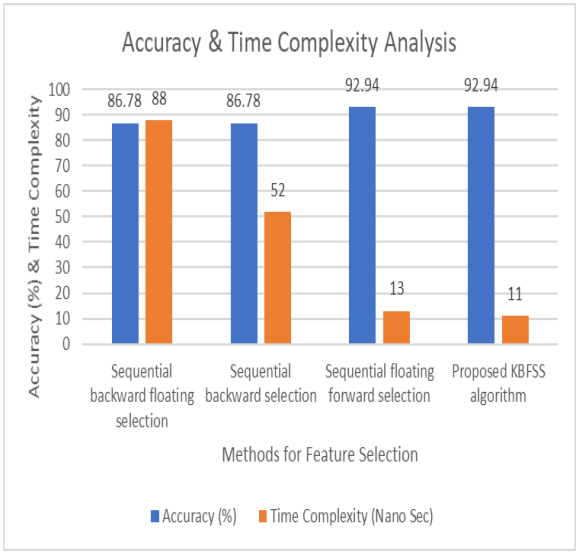

Fig. 7 Accuracy & Feature Selection Analysis

Furthermore, the proposed method also incorporates the dynamic rule generation algorithm for detecting the intrusions. Henceforth, the combined accuracy is also analysed here [Table – 14].

TABLE XIV: COMPARATIVE ANALYSIS – ACCURACY

Method	Accuracy (%)
Sequential backward floating selection	86.78
Sequential backward selection	86.78
Sequential floating forward selection	92.94
Proposed KBFSS + DIREID algorithm	93.24

The results are also analysed visually [Fig – 8].

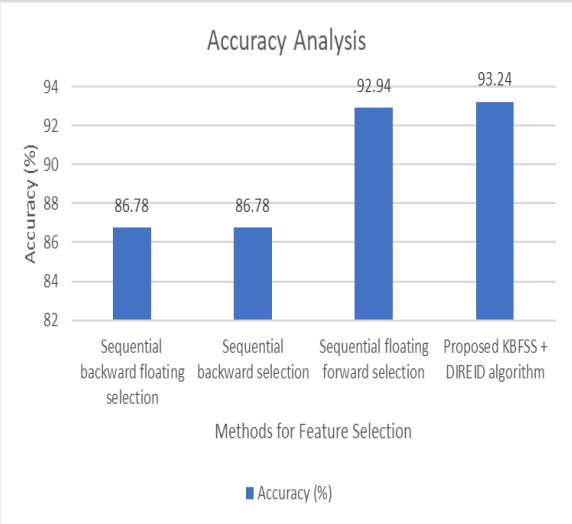

Fig. 8 Composite Accuracy Analysis

Hence, it is natural to realize that, the proposed hybrid algorithm has clearly outperformed the other existing popular method.

IX. CONCLUSIONS

The effective intrusion detection in the era of modern computing must accomplish the detection of intrusion not only on the traditional networks, rather determine the intrusion scenarios on data processing framework as most of the traditional algorithms are migrated to the data processing frameworks so as the networks. This work identifies the gap in the existing researches and realizes that the nature of the attacks has changed and due to the mapper – reducer architecture of the data processing frameworks, the newer attacks are getting introduced every day. Hence this work addresses the demand of dynamic rule-based intrusion detection algorithm for detecting the existing and any new intrusions. During the course of research, this work also identifies the need for reducing the dimension of the features relying on which the intrusions can be detected as on the data processing framework, processing capabilities are highly crucial and reducing the feature set size can be highly beneficial. Thus, this work also proposes a novel knowledge-based feature selection algorithm. The combined effort by the two proposed algorithms have demonstrated significant performance improvements in terms of increased accuracy and reduced time complexity for making the world of data processing architectures more secure.

REFERENCES

1. A. Sultana, M. A. Jabbar, "Intelligent network intrusion detection system using data mining techniques", Proc. IEEE 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT), pp. 329-333, Jul. 2016.
2. L. Oneto, F. Bisio, E. Cambria, D. Anguita, "Statistical learning theory and ELM for big social data analysis", IEEE Comput. Intell. Mag., vol. 11, pp. 45-55, Aug. 2016.
3. A. Chaudhary, V. Tiwari, A. Kumar, "A novel intrusion detection system for ad hoc flooding attack using fuzzy logic in mobile ad hoc networks", Proc. IEEE Recent Adv. Innov. Eng. (ICRAIE), pp. 1-4, May 2014.
4. S. Malhotra, V. Bali, K. K. Paliwal, "Genetic programming and k-nearest neighbour classifier based intrusion detection model ", Proc. IEEE 7th Int. Conf. Cloud Comput. Data Sci. Amp Eng. Conf., pp. 42-46, Jan. 2017.
5. R. Sen, M. Chattopadhyay, N. Sen, "An efficient approach to develop an intrusion detection system based on multi layer backpropagation neural network algorithm: IDS using BPNN algorithm", Proc. ACM SIGMIS Conf. Comput. People Res., pp. 105-108, 2015.
6. M. Tabatabaefar, M. Miriestahbanati, J.-C. Grégoire, "Network intrusion detection through artificial immune system", Proc. Annu. IEEE Int. Syst. Conf. (SysCon), pp. 1-6, Apr. 2017.
7. T. Mehmood, H. B. M. Rais, "SVM for network anomaly detection using ACO feature subset", Proc. IEEE Int. Symp. Math. Sci. Comput. Res. (iSMSC), pp. 121-126, May 2015.
8. M. S. Pervez, D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs", Proc. IEEE 8th Int. Conf. Softw. Knowl. Inf. Manage. Appl. (SKIMA), pp. 1-6, Dec. 2014.
9. Y. Guang, N. Min, "Anomaly intrusion detection based on wavelet kernel LS-SVM", Proc. IEEE 3rd Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT), pp. 434-437, Oct. 2013.
10. T. Yerong, S. Sai, X. Ke, L. Zhe, "Intrusion detection based on support vector machine using heuristic genetic algorithm", Proc. IEEE 4th Int. Conf. Commun. Syst. Netw. Technol. (CSNT), pp. 681-684, Apr. 2014.
11. Z. Chen, T. Lin, N. Tang, X. Xia, "A parallel genetic algorithm based feature selection and parameter optimization for support vector machine", Sci. Program., vol. 2016, Jun. 2016.
12. K. S. Desale, R. Ade, "Genetic algorithm based feature selection approach for effective intrusion detection system", Proc. IEEE Int. Conf. Comput. Commun. Inform. (ICCCI), pp. 1-6, Jan. 2015.
13. The Economist, Nov 2011, "Drowning in numbers – Digital data will flood the planet and help us understand it better",http://www.economist.com/blogs/dailychart/2011/11/big-data-0
14. Zhikui Chen, Fangming Zhong, Framework of integrated data processing : A review, IEEE International Conference on Data processing Analysis (ICBDA), 2016
15. N. Chand, P. Mishra, C. R. Krishna, E. S. Pilli, M. C. Govil, "A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection", Proc. IEEE Int. Conf. Adv. Comput. Commun. Amp; Autom. (ICACCA) (Spring), pp. 1-6, Apr. 2016.
16. R. Vijayanand, D. Devaraj, B. Kannapiran, "Support vector machine based intrusion detection system with reduced input features for advanced metering infrastructure of smart grid", Proc. IEEE 4th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS), pp. 1-7, Jan. 2017.
17. M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.