Embedded Feature Selection Method for a Network-Level Behavioural Analysis Detection Model

Mohammad Hafiz Mohd Yusof [1,2], Mohd Rosmadi Mokhtar[2], Abdullah Mohd. Zain[2], Carsten Maple[3]
[1] Faculty of Information Technology & Sciences, INTI International University, Nilai, Negeri Sembilan, 71800, Malaysia
[2] Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, 43600, Malaysia
[3]Cyber Security Center, WMG, University of Warwick Coventry, CV4 7AL, U.K

***Abstract*—Feature selection in network-level behavioural analysis studies is used to represent the network datasets of a monitored space. However, recent studies have shown that current behavioural analysis methods at the network-level have several issues. The reduction of millions of instances, disregarded parameters, removed similarities of most of the traffic flows to reduce information noise, insufficient number of optimised features and ignore instances which are not an entity are amongst the other issue that have been identified as the main issues contributing to the inability to predict zero-day attacks. Therefore, this paper aims to select the optimal features that will improve the prediction and behavioural analysis. The training dataset will be trained to use the embedded feature selection method which incorporates both the filter and wrapper method. Correlation coefficient, *r* and weighted score, w_j will be used. The accepted or selected features will be optimised uses Beta distribution functions, *β*, to find its maximum likelihood, l_{max}. The final selected features will be trained by the Bayesian Network classifier and tested through several testing datasets. Finally, this method was compared to several other feature selection methods. Final results show the proposed selection method's performance against other datasets consistently outperform other methods.**

Keywords—Feature selection; intrusion detection; behavioural analysis

I. Introduction

Behavioural analysis has become a trending research area compared to signature-based studies [1]. Computer networks in general are less studied due to the lack of leveraging behaviour of malware attacks in the network environment [2]. An article by [3] stated that behavioural-based detection methods are effective in malware detection and prediction. Meanwhile, the author of [4] describes the behavioural analysis model as being used to discover malware adaptation tactics that are difficult to understand through static signatures. These statements have led to the discussion in this paper on the existing studies in relation to behavioural-based analysis methods, specifically in the network environment.

Features selected could be different between research field like in image authentication [19] steganography [20] and wireless sensor networks [18]. Feature selection in network-level behavioural analysis studies is used to represent the network dataset of a monitored space [4]. The research of [4] used Internet Protocol addresses as a feature to represent the monitored space. The author in [5], on the other hand, used application protocol HTTP to represent his selection feature. However, recent studies have shown that current behavioural analysis methods at the network-level have several issues, such as the inability to predict zero-day attacks, high-level assumptions, non-inferential analysis, a lack of ground truth datasets, a lack of distribution modelling refinement processes and performance issues [6]. Feature selection methods give a better understanding of the dataset, prepare a framework or technique to improve prediction performance, reduce computational time, reduce the effect of dimensionality and improve prediction performance in machine learning or in pattern recognition applications [7].

However, network features are different, whereby the packets are too discrete and robust, and might therefore not be sufficiently modelled through the time of propagation. To improve the accuracy of the dataset, a certain elimination algorithm has to be applied. Removing information or instances from the network dataset will lead to inaccurate results [6]. Since suitable algorithms for extracting portions of the feature from the packets automatically is an open question [8] in research, this paper aims to select for the **optimal features** that will improve the prediction and behavioural analysis.

II. Preliminaries

Based on the above-mentioned issues, three problems are the inability to predict, high-level assumptions and non-inferential analysis. These could be further grouped into their mutually shared common criteria, summarised in the following points.

A. *Reduced Parameters (Instances), θ*

The numerical characteristics of a population are often denoted by parameter θ and the numerical description of a subset is denoted by y which is uncertain before a dataset is obtained. The level of uncertainty decreases once the dataset has been identified. Given space, Θ is set of potential parameters θ, thus $\theta \in \Theta$ (2) so that the product of all possible outcomes of parameter Θ and unknown parameters X becomes Ω denotes the universal, Ω=X. Θ [9], thus it is important to obtain as much information about the parameters as possible to derive informative results.

As illustrated in Figure 1, conceptually, these are the building blocks of a universal set Ω which is the outcome of all possible parameters and the unknown parameters as well. Given Θ is the space of all possible parameter values θ where $\theta \in \Theta$. In the diagram, there are two sets of parameters θ_n and

θ_{an}. These sets are the element of the parameter space Θ. If a method is used to reduce or discard each of the parameter sets, it will limit the parameter or instance information which could be used to drawn further conclusions or the inability to predict unknown (zero days) attacks. This problem could be solved through the prior information.

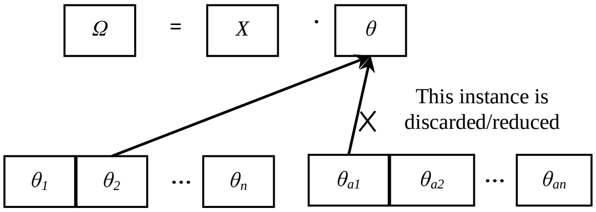

Fig. 1. Reduced Parameters or Instances Explained in Diagram.

B. Lack of Priori, $p(\theta)$

Prior distribution $p(\theta)$ explains the certainty that θ signifies the accurate population characteristics explained in [9]. As shown in Figure 2 above, it is a derivation of the previous problem. As far as the previous problem is concerned, the parameters are reduced by some reduction process or totally ignored, which could affect the results or conclusions. It was also stated earlier that the problem could be resolved by establishing prior information. The prior information, or simply priori, is done by the probabilistic method. For instance, parameters θ_n and $\theta_{an,}$ instead of being reduced or limited, have been represented by $p(\theta_n)$ and $p(\theta_{an})$ which is the notation for prior information. However, this doesn't happen in the previous method. Instead, they choose the method which ignored prior information like an in state-transition or another method that represents the collection of information of the main features in the data collection without determining inferential or in-depth analysis. It can also involve simply assuming the probability of the parameter occurrences. This leads to high-level assumptions and non-inferential analysis problems.

In a volatile or in a critical infrastructure network environment such as in the energy industry, the lack of prior information could cause a catastrophic false alarm as happened in the history of Iranian nuclear plant, in a Saudi Aramco oil and gas plant and in the healthcare industry. A lack of information capabilities could lead to the breach of patient information and malware attacks as happened during the WannaCry malware that attacked hospitals, mostly in Europe.

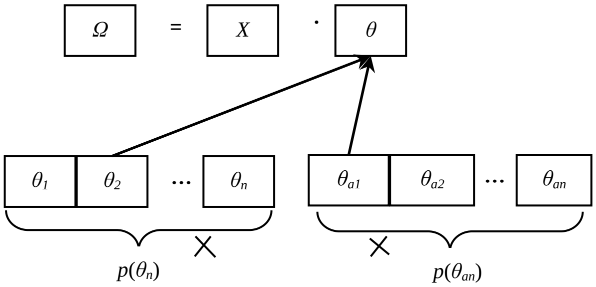

Fig. 2. Lack of Priori Information Explained in Diagram.

Due to the **reduced instances issue and data normalisation practices,** and since the malware analysis in computer networks in general are less studied due to the lack of leveraging behaviour of the malware attacks in the network environment as mentioned by the author in [2], this paper has proposed a feature selection method and process flow that suits the complexity and traffic in networks which use the embedded feature selection method and optimised beta distribution function.

III. Feature Selection

Normally, feature selection starts with a pre-processing phase like for instance in the work by [10]. They had their dataset processed early for it to be later represented it in a vector of real numbers. Then the data was normalised, selected and classified. The data acquired through the data collection stage was firstly analysed to produce the elementary instances or features. The whole processes contained three main stages which were pre-processing, feature selection and classification. This payload was loaded for pre-processing. The data was presented as a real number vector. Thus, every symbolic feature or nominal feature was converted into numerals. In the NSL dataset, the nominal feature included a protocol of type UDP, TCP and ICMP and the service protocol of type FTP, HTTP and telnet.

The dataset was then analysed using feature selection and finally classified by the chosen classification method. The feature selection method gives us a better understanding of the dataset, prepares a framework or technique to improve **prediction performance**, reduces computational time, reduces the effect of **dimensionality** and improves the prediction performance in pattern recognition or machine learning [7].

As defined in [11], feature selection methods were classified into wrapper and filter methods. Finally, the embedded methods combine both the filter and wrapper method, and include feature extraction by means of integral phase of the training procedure deprived of disjointing the dataset into testing datasets or training datasets. On this research, we will apply the proposed feature selection method on a supervised dataset.

A. Filter Method

The filter method uses variable ranking methods as the basis for its variable selection criteria. The ranking method is used mainly because of its uncomplicatedness, simplistic approach and its success history in recording certain pragmatic applications. Unique features contain useful and relevant information about the property of the dataset or instances [7]. This relevant and useful property is used to measure [11] the usefulness of the feature compared to other feature, and finally to discriminate it from that other feature's label. For instance, one of the criteria of the simplest principle is the correlation coefficient, also known as Pearson correlation. Correlation coefficient ranking is able to identify linear dependencies between the target and the variable. The Pearson **correlation coefficient**, *r* is defined as below.

$$r = \frac{1}{n-1}\Sigma\left(\frac{x_i-\mu_x}{\sigma_x}\right)\left(\frac{y_i-\mu_x}{\sigma_y}\right) \quad (1)$$

Where *n* signifies the full training set number whereby set, x_i indicates the *i*th variable of feature *x*. Meanwhile σ_x and μ_x are the standard deviation and mean aimed at feature *x* respectively. Whilst y_i could be the label or other correlated features or to test dependency features in the dataset whereas σ_y and μ_y are the standard deviation and mean aimed at feature *y* respectively.

The above equation is to determine the product of *z*-value of both features. *Z*-value is to determine how a single instance of a set of features is positioned from its mean and its standard deviation. So any $-1 \leq r \leq 1$ value that is drawn towards positive 1, we could conclude that there is positive correlation.

Network traffic are usually **linearly dependent** on each other as discussed by [8], such that some novel cyber or network attacks are variations of the previous identified attacks and its signature could be sufficient to detect and prevent some other novel variants. Coefficient correlation is suitable for processing network flow. They also explained that **some probing attack scans are correlated** with a much larger time scanning interval compared to normal traffic. However, the correlation ranking is only able to detect linear dependencies between the target and the variation.

B. Wrapper Method

Unlike filter methods (FM) which use the ranking method as the criterion for its relevance feature, the wrapper method (WM) on the other hand relies on the classifier or **classification method** for obtaining a feature or instance subset.

Therefore, the simplified version of algorithms for instance, sequential searching algorithm or evolutionary algorithm such as Genetic Algorithm (GA) or Particle Swarm Optimization (PSO) that will harvest local optimum outcomes. They are applied as they can generate good computationally feasible results.

Wrapper methods can be divided into Heuristic Search Algorithms and Sequential Selection Algorithms. Sequential Selection Algorithm are named as it is because its algorithm is designed as iterative in process. It starts with a full dataset and in the process, the features are removed till the maximum objection function has been gained. On the other hand, it begins with an empty set and throughout the process, the features are added until they reach the maximum.

On the other hand, heuristic search algorithm is about reaching the local optimum results by applying an evolutionary algorithm such as a Genetic Algorithm (GA). GA (31) is used to select the features whereby the chromosome bits are used to denote the selected features. It is based on the natural selection theory by Darwin. Searching the GA provides both data exploration and data exploitation.

$$f = \{\sum_{i=1}^{n} c_i v_i \text{ , if} \sum_{i=1}^{n} c_i w_i \leq k \text{ . } 0 \ otherwise \quad (2)$$

Particle Swarm Optimisation (PSO) assumes a “swarm” of *N* particles (32). General Particle Swarm Optimization algorithm is simple. PSO is initialised with a group of random solutions or particles, which is then searched for by learning of the next generations. Particles will swarm throughout the space, and are tested or evaluated across the fitness criterion. In each iteration, the particles will be updated by following two “best” values. The below equation represents the PSO algorithm.

$$v_{n+1} = v_n + c_1 rand1(\,) * \left(p_{best,n} - currentPosition_n\right) + c_2 rand2(\,) * g_{best,n} - \ currentPosition_n \quad (3)$$

Where, v_{n+1} is the velocity of particle at the *n+1*th iteration and v_n is the velocity of particle at *n*th iteration. C_1 is acceleration factor related to the *gbest* and C_2 is the acceleration factor related to *lbest*. *rand1*( ) and *rand2*( ) is the random number between 0 and 1.

The main disadvantage of a wrapper method is that it requires a number of computational processes in order to obtain the final feature. Having said that, for instance, if the dataset sample is large, then most of the execution of the algorithm will be allocated to train the predictor. Note that our research will reduce this by calculating the optimised value in the ranking process. In the next section, this paper will elaborate on the embedded methods to then try to leverage the drawbacks or disadvantages found in the Wrapper or Filter methods.

C. Embedded Method

The main purpose of the embedded methods [12,13,14] is to lessen the computational time that is used to reclassify the dissimilar subsets which completed in WM. This is done by incorporating the feature selection process in the FM as part of the training process [7]. The main approach is to incorporate FM and WM.

For instance, a method was to use the weights of a classifier to remove the feature based on the rank [12,15]. For example, let w_j be denoted as

$$w_j = \frac{\mu_j(+) - \mu_j(-)}{\sigma_j(+) + \sigma_j(-)} \quad (4)$$

Where μ_j (-) and μ_j (+) and are the mean of samples in class + and class – and σ_j is the variance of the respective classes and *j*=1 to D. Equation 13 can be used as a ranking criterion to sort the features. The rank vector *w* can be used to classify since the features rank proportionally. This contributes to the correlation. Another weighted score is the true normal score, whereby in order to create a normal profile, it is necessary to index each attributes’ instances as *i*=1,2…*n*. The model was build based on the ratio of the normal number of training data, R_i against the total number of packets associated with each attribute, N_i. The probability of the normal score, P_i=R_i/N_i is represented by

$$Pi = \sum_{i=1}^{n} \frac{R_i}{N_i}, i = 1,2,3 \dots, n \quad (5)$$

IV. Dataset

Table 1 shows the basic features of Network Socket Layer (NSL) dataset which is an updated version of the KDD Cup 1999 data set. The KDD Cup 1999 dataset was used for a data mining completion which was organised in conjunction with the Fifth International Conference on Knowledge Discovery and Data Mining. During the competition, the challenge was to design a **predictive model** or network **intrusion detector** or that was able of differentiating between attack connections or intrusions, and baseline connections. This dataset contained standard data to be analysed, which also included varieties of computer-generated intrusion scenarios in a military network environment, specifically simulating LAN connectivity of U.S Air Force [8].

However, from a network practitioner's point of view, the KDN or NSL datasets are **not realistic** and do not reflect modern attacks, and not even attacks back in 1998 [16]. Today's attacks are primarily SQL injections. The KDN dataset was also focused around attacks with some background noise, while the actual traffic was largely data. Furthermore, it was a simulated dataset within a large virtual network.

To apply objectivity, in this research, final classification method using Bayesian Network will be applied over a ground-truth dataset. That ground-truth dataset or simply raw dataset was obtained from the local asset, which was a host tagged to among the largest healthcare provider in Malaysia. This is more adequate to strategize the scan rate of one-to-one modelling. The traffic is more resemblance to one-to-one connection. One to one model is to mimic a connection of a single infected machine that is transacted throughout the network.

TABLE I. BASIC FEATURES OF INDIVIDUAL TCP CONNECTIONS

Feature Name	Description	Type
duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	discrete
service	network service on the destination, e.g., http, telnet, etc.	discrete
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
flag	normal or error status of the connection	discrete
land	1 if connection is from/to the same host/port; 0 otherwise	discrete
wrong_fragment	number of ``wrong" fragments	continuous
urgent	number of urgent packets	continuous
duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	discrete
service	network service on the destination, e.g., http, telnet, etc.	discrete
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
flag	normal or error status of the connection	discrete

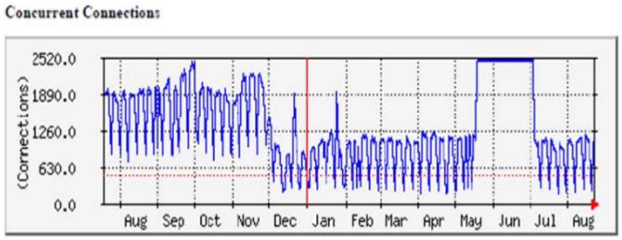

Fig. 3. Ground-Truth Dataset from Asset (Internet Load Balancer) that is Tagged to the Healthcare Provider in Malaysia.

Traffic captured and monitor from August 2016 until August 2017 and traffic information was extracted from the Cisco FMC, Steal Head Riverbed WAN optimizer and Hgiga internet load balancer network appliances as [4] stated that network activity profiles of infection network environment is dependent on both distribution of activity across internet and malware propagation techniques or targets that might differ from each network population profiles.

Figure 3 below manifests definition laid by [4] which indicates traffic distribution activity across the internet could potentially highlight the malware propagation activity. It shows some scanning activity happened from early May 2017 until end of June 2017 and after that period the counts appeared to decline to a baseline level towards the end of data collection period.

IT personnel from the healthcare provider confirmed that during that period Trend Micro DDAN (Virtual Analyzer) has sent a lot of suspicious object (SO) information to the Trend Micro OS indicated some malicious activity and Trend Micro CM has pushed latest signature to all the endpoints to disinfect the malware attacks.

Evaluation to demonstrate that malware really propagated during this period from May 2017 to July 2017 is presented in the following subsection report. The healthcare equipped with Trend Micro DDAN report which trapped all suspicious object in the environment and analyze it in their Virtual Analyzer (VA).

V. METHODOLOGY

Figure 5 depicted the whole proposed method process flow was based on work by [10], whereby the training set was first processed in the pre-processing phase. In [10], the pre-processing was conducted to differentiate between normal and attack traffic. The data was then weighted by normal function whereby the nominal data was converted into numeric data, followed by calculating its normal function. The distribution of the numerical data was determined by its normal function. The whole process is known as data normalisation. Instances will be greatly reduced during this process and it affects the classification process (detection or prediction), as proven in the results and discussion section. In the proposed method, pre-processing was done by applying equation whereby, each features' instances were converted into its normal score and normal traffic was compared to produce a **baseline** value and attack traffic. This was compared to produce a **threshold** value.

The next phase is followed by the feature selection process. In [10], the feature selection used flexible MI, which was suggested to select the feature by argmax. In the proposed feature selection method, equation 3 (***Eq.*** 3) which is for numerical instances and equation 13 (***Eq.*** 13) which is for nominal instances, was applied and the instances that can be manageably optimised by the modelled optimisation function and distributed by the generic Beta function, β of which its maximum likelihood, l value will be selected. The results show that by processing the instances optimisation value and its beta distribution function, the features selected are significantly minimum yet the detection accuracy is very high compared to the previous method. The false alarm rate has also been reduced.

Phase 1: Pre-processing modelling

B, packet capture or space for baseline traffic and

B=b= $\{b_1, b_2, \ldots b_n\}$ (6)

C, packet capture or space of attack traffic

C=c= $\{c_1, c_2, \ldots c_n\}$ (7)

The above is the representation of the raw dataset for both the attack and baseline traffic. Equation (6) and equation (7) were applied in equation (8), which produced a new equation.

$$Pb, c = \sum_{b,c=1}^{n} \frac{R_{b,c}}{N_{b,c}}, b, c = 1,2,3 \ldots, n \quad (8)bc$$

Where $P_{b,c}$ is the normal score for dataset b and c. The output from this process is the dataset that will be labelled as numerical or nominal and a change of notation for instance f_1 to indicate feature number 1. Equation 8*b,c* will be used to represent the nominal data for future feature selection processes.

Phase 2: Embedded feature selection modelling

The embedded method incorporated both the Filter (FM) and Wrapper method (WM). As in the work done by [10], the selected FM method used a correlation coefficient, *r* which is good to process multidimensional data with multi-array instances. In this work, it was used to process the numerical types of dataset. The formula given is:

$$r_{b,c} = \frac{1}{n-1} \Sigma \left(\frac{x_{bi} - \mu_{xb}}{\sigma_{xb}}\right)\left(\frac{y_{ci} - \mu_{xb}}{\sigma_{yc}}\right) \quad (1)bc$$

Whereby r_{bc} is the coefficient value for both dataset *b* and *c*. Equation (1)*b,c* is formulated by applying equation (4) and (5) into equation (1). In [10] Any r<0 will be rejected. However, in the proposed method, any **r>0** will be rejected. Then for nominal dataset, weighted score, w_j will be used for feature selection processing. Any w_j<0 will be rejected. Then the selected features will be optimised to avoid optimVal (optimal value) errors in the Beta distribution function in order to find the likelihood value. The optimisation formula has been given below.

Baseline_f_i_beta =

$10^{-x} \sum i$=1 to n , (-) $\log_{10}$ $Baseline_{fibeta}$ (9)

Where -*x* is the power value for absolute value 10, which is to avoid the instances exceeding 1.0 which could produce optimVal error and *m* must be between 0<*m*<1. Beta function, β is given by the formula below.

Beta ~ (λ_{fi} ; α, β)=

$$\frac{1}{Beta\,(\alpha,\beta)} . \lambda_{fi}{}^{\alpha-1}(1 - \lambda_{fi}).^{\beta-1} \;, where\; 0 < \lambda_{fi} < 1 \quad (10)$$

Whereas maximum likelihood, l function in the form of log function is given by the formula below.

Maximum likelihood, ℓ_{fi_beta}

$= \ln [\sum_{fi=0}^{n} \frac{1}{Beta(\alpha,\beta)} . \lambda_{\mathrm{fi}}{}^{\alpha\text{-}1} (1\text{-} \lambda_{\mathrm{fi}})^{\beta\text{-}1}$,

where

$0 < \lambda_{fi} < 1]$=

$(\alpha\text{-}1) \sum_{fi=0}^{n} \ln(fi) + (\beta\ \text{-}1) \sum_{fi=0}^{n} \ln(1\text{-}fi)\text{-}N \ln\mathrm{Beta}(\alpha, \beta)$ (11)

Where *N*, is the total number of i.i.d observations

Phase 3: Classification modelling

This classification was based on the work by [4] on Naïve Bayes. However, we modified it to **incorporate the Bayesian Network approach**. Bayesian Network as depicted in Figure 4, is approach that were classified under probabilistic theorem and being highly chosen is mainly due to their flexibility and ability to model uncertain events such as the Bayes theorem which has been considered as the state-of-the-art technology [6]. Because of the intuitive ability to model uncertainty and complex chronological relationships amongst variables, Bayesian network is successfully applied in several research areas and domains [17]. State-of-the-art predictive analytics method of uncertainty and the detection of the unknown, using the Bayesian Network method, have been proven in other research areas, especially in the domain of Clinical Expert System studies, Artificial Intelligence (AI) and Pattern Recognition. Modelling the classification based on Bayesian Network has been given below.

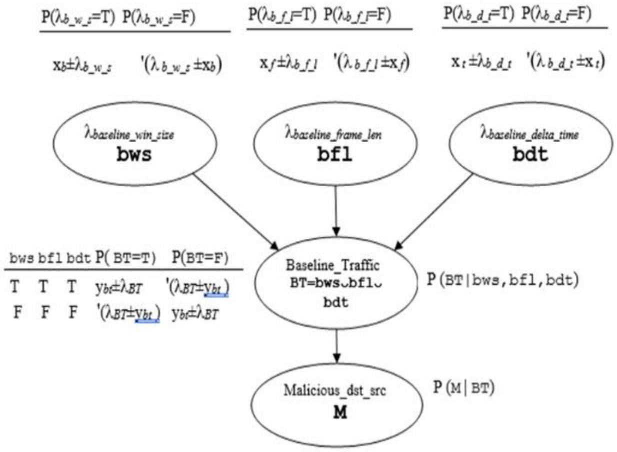

Fig. 4. Bayesian Network Classification Model.

The model is then written in its conditional probability as derived in the following forms. This model is supplied to the classifier.

$$Pr(f_i) = \lambda_{fi} \text{ where } i=1\ldots.j\ ,\ 1\ldots.k\ ,\ 1\ldots.l \quad (12)$$

$$Pr(f_i \ldots j,k,l) = Pr(\lambda_{fi}\ldots j).\ Pr(\lambda_{fi}\ldots k).\ Pr(\lambda_{fi}\ldots l) \quad (13)$$

$$Pr(Malicious,\ M) = Pr(\lambda M).\ Pr(\text{‘}\lambda M) \quad (14)$$

$$Pr(f_i\ldots j,k,l|\ f_i) = Pr(\lambda_{fi}\ldots j.\ \lambda_{fi}).\ Pr(\text{‘}\lambda_{fi}\ldots j.\ \lambda_{fi}) \quad (15)$$

$$Pr(M\ |\ f_i\ldots j,k,l|\ f_i\ )= \lambda_M\ .\ \lambda_{fi}\ldots j\ .\ \lambda_{fi}$$

Equation. (13) in *Equation.* (14)

VI. Results and Discussions

In this section, we have discussed the results of applying the proposed Embedded Feature Selection model to the KDD dataset using the Bayesian Network classifier. We implemented two feature selection methods as shown in phase 2 in the methodology section, whereby the features training dataset and variations of NSL and also the KDD dataset in the later operation was used in the testing dataset as well. Comparisons between the selection algorithm could only be done using a single dataset and the selection techniques indicated that more feature or instance information is not always good in the context of machine learning applications [7]. Table 2 below shows the training dataset descriptions.

TABLE II. Training Dataset Descriptions

Type of packet	Number of packet	Total packet
Normal	9711	22544
Attack	12834	22544

A. Feature Selection Results

The features selected were still on 1-dimensional with 2 arrays of data. This means that the feature is the same attribute but constructed in 2 arrays of information.

For duration, *f1* correlation coefficient, the r_{f1} score was -0.009742335. This indicates a negative correlation. Correlation ranking can only detect linear dependencies between the variable and the target. Hence, in the case any $-1 \leq r \leq 1$ value that is drawn towards positive 1, we can conclude that there is a positive correlation. Negative correlation means that there is no linear dependency between the two datasets. Thus, the duration of traffic transactions between normal and attack traffic has no correlation and no dependency. In this case, $r<0$ will be rejected. Note, however, that in some models, it will be accepted as no correlation, which means that a threshold could be constructed.

The results of the entire feature selection process have been summarised in the following table 3.

TABLE III. Feature Selection Process Summary

Features	Correlation score, r_{fi}	Weighted score, w_{fi}
f_1	-0.009742335	-
f_2	0.976564	-
f_3	0.026048497	-
f_4	0.781732335	-
f_5	-0.000713582	0.019534
f_6	0.002013838	0.098281
f_7	-	0.023361
f_8	-0.00419	0.028317
f_9	-	0.025794

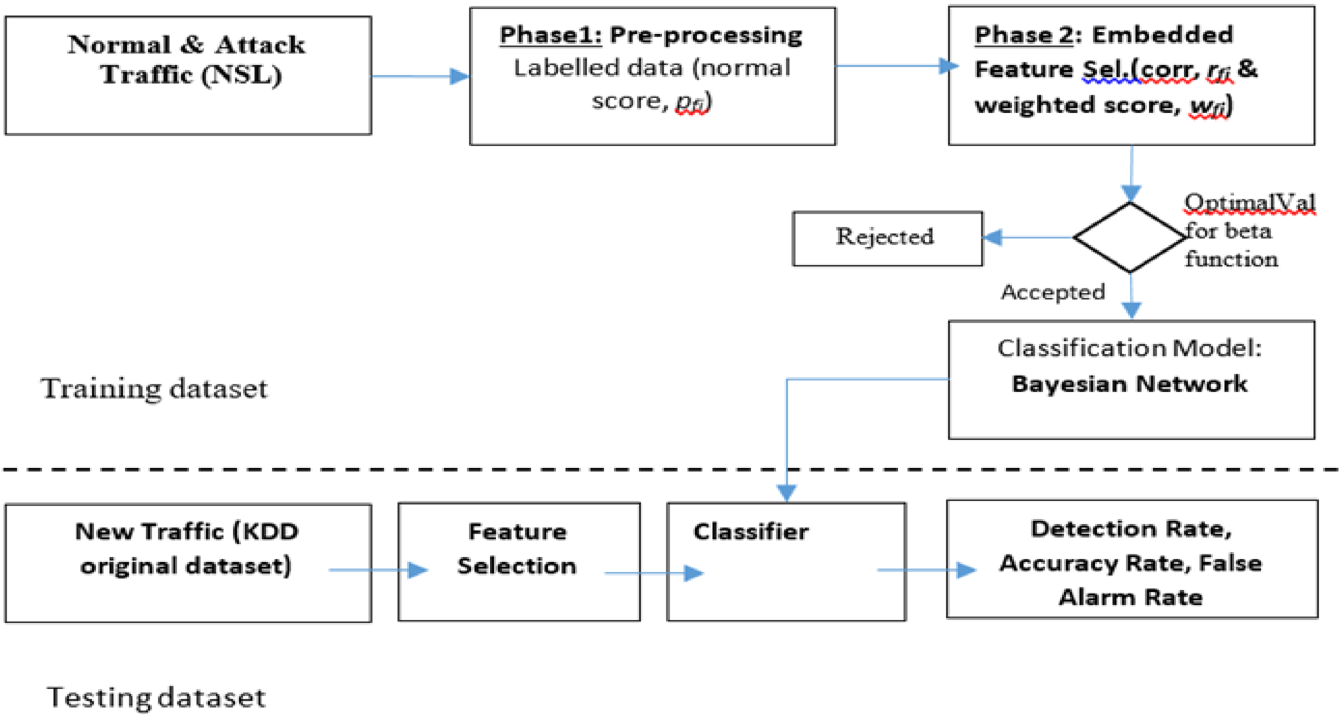
Normal & Attack Traffic (NSL)
Phase1: Pre-processing Labelled data (normal score, ρfi)
Phase 2: Embedded Feature Sel.(corr, rfi & weighted score, wfi)
OptimalVal for beta function
Rejected
Accepted
Classification Model: Bayesian Network
Training dataset
New Traffic (KDD original dataset)
Feature Selection
Classifier
Detection Rate, Accuracy Rate, False Alarm Rate
Testing dataset

Fig. 5. Proposed Feature Selection Workflow.

TABLE IV. FEATURES ACCEPTED OR REJECTED DURING FEATURE SELECTION PROCESS

Feature selection approach	Number of features	Features Selected
Original Features	$f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9$	-
Weighted score *w*>0	$f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9$	f_5, f_6, f_7, f_8, f_9
Weighted score *w*<0	$f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9$	-
Correlation coeff. *r*>0,	$f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9$	f_2, f_3, f_4, f_6
Correlation coeff. *r*<0	$f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9$	f_1, f_5, f_8
Maximum likelihood	$f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9$	f_1, f_5, f_6

Finally, based on the feature selection model, table 4 below shows the list of features that were selected or rejected during the process.

B. Classification Rresults

Only the following features in table 5 below were selected after optimisation, distributed using the beta function. The maximum likelihood features will be selected.

The performance of this selected feature over its classification model was based on the true positive (TP value), true negative (TN value), false positive (FP), false negative (FN), detection rate (DR), accuracy (ACCR) and false alarm rate (FAR).

Detection rate, on the other hand, is used to measure true positive traffic over the sum of true positive and false negative (positive traffic wrongly classified as negative). The formula is the following

$$Detection\ Rate, DR = \frac{TP}{TP+FN} \quad (16)$$

Accuracy is used to measure all true traffic which consists of the sum of the true positive and true negative over the sum of all traffic of a true positive, true negative, false positive and false negative nature. The formula is denoted as the following.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (17)$$

TABLE V. OPTIMISED FEATURES

λ*duration , f1*	Shape 1: 0.3601539 Shape 2: 533.8212008 Loglikelihood:66584.47	Mean (optimized prior), $\lambda_{duration} = \frac{\alpha}{\alpha+\beta}$ = 0.000674
λ*src_bytes , f5*	Shape 1: 0.5537134 Shape 2: 1347.6133442 Loglikelihood: 67446.38	Mean (optimized prior), $\lambda_{src\_bytes} = \frac{\alpha}{\alpha+\beta}$ = 0.000411
λ*dst_bytes, f6*	Shape 1: 0.7658164 Shape 2: 1466.8806705 Loglikelihood:63927.07	Mean (optimized prior), $\lambda_{dst\_bytes} = \frac{\alpha}{\alpha+\beta}$ = 0.000522

TABLE VI. KDDTRAIN+_20PERCENT CLASSIFICATION TABLE

Lamda information	Baseline	KDDTest+_20Percent	Ratio (dataset over attack)	Threshold (spike/attack)	Threshold (ratio normal over attack)	Difference
Pr(λduration)	47.07	4507	0.351	12833	0.00366	0.347
Pr(λsrc_bytes)	2530.77	10973.087	0.671259475	16347.0134	0.15481562	0.516443855
Pr(λ dst_bytes)	4165.553553	957.895274	2.082316478	460.0142601	9.055270487	-6.972954009
Pr(*BT*)	496249242.3	47373387043	0.490903953	96502354064	0.005142354	0.485761599
Pr(*BT*|λ duration)	23360051486	2.13512E+14	0.172407396	1.23841E+15	0.0000188629	0.17238853 32
Pr(*BT*|λ src_bytes)	49625177 3.1	5.19832E+14	0.32952393	1.57753E+15	0.0000003146	0.3295236151
Pr(*BT*|λ dst_bytes)	2.06715E+12	4.53787E+13	1.02221739	4.43925E+13	0.0465654041	-0.9756519855

Finally, the false alarm rate (FAR) is used to measure the false positive alarm, which means the negative traffic that was wrongly classified as positive. This is a very serious issue because it may cause an attack vector. The formula is denoted as the following.

$$False\ Alarm\ Rate, FAR = \frac{FP}{FP+TN} \quad (18)$$

The above table 6 shows the differences between the two dimensional information of each features' **lambda, λ information**. It is generated from the differences between the ratio value of the testing dataset, in this case KDDTest+_20Percent, and attack traffic over the ratio of the normal dataset over attack traffic.

For NSL-40% dataset, *Pr*(λ duration) training ratio exceeded the baseline threshold by 0.699, which indicates that this is attack traffic. *Pr*(λduration) differences, this time, had increased almost 50% from the previous dataset. This may be due to a 20% increase in the traffic. Out of that, only 0.04% of this traffic was flagged as normal. Thus, the entire dataset was still attack traffic and was flagged as negative tuple or *TN*, the same as in the previous dataset. It was then a true alarm or *TP* because the alarm truly reflected the tuple condition.

For the train dataset, the *Pr*(λ duration) training ratio, this time, never exceeded the baseline threshold. It scored below the threshold by -0.000305238, which indicates that this is a normal traffic. Thus, the entire tuple will be flagged as positive tuple or *TP*. It will then be alarmed as a true alarm or *TP* because the alarm truly reflected the tuple condition, as a true positive.

TABLE VII. PROPOSED METHOD PERFORMANCE AGAINST OTHER DATASET

Dataset	Desc.	Detection Rate	Accuracy	False Alarm Rate (FAR)
NSL-40%	Attack flow (0.04% normal)	**100%** (1.0)	**86%** (0.857143)	**14%** (0.142857)
KDDTest+_20Percent	Attack flow	**100%** (1.0)	**86%** (0.857143)	**14%** (0.142857)
KDD-Train+	Normal flow	**86%** (0.857143)	**86%** (0.857143)	**0%**

Table 7 shows above the proposed method performance against other dataset. For instance, NSL-40%, the detection rate was 100% because the classification model successfully classified all alarms as attack traffic even though 0.04% of the flow was normal traffic. Only one tuple *Pr*(λdst_bytes) was flagged as positive which is correct, however the intersection probability *Pr*(*BT*|λdst_bytes) was actually negative. This is a false positive alarm, whereby negative traffic was alarmed positive. Hence, it affects the *FAR* and accuracy as well, which scored 14% and 86% respectively. This is a serious failure, however, due the intersection probability that is included in this model, this flag could be re-examined and re-flagged to the correct alarm.

Finally, table 8 above shows proposed method performance against other feature selection method. Example, for the method that uses correlation coefficient, r whereby the accepted feature selection was when *r*>0, most features were nominal features. The features need to be changed into a numerical dataset and afterwards, distributed using data normalisation. When this happens, most of the instances of the features will be altered and reduced in dimension or volume. Thus, can be seen the poor result obtained, especially in relation to the false alarm, whereby 89% of the detections were false alarms. The feature selection process was validated 5 folds.

Then we apply the classification model to predict zero - day attack in the ground truth dataset mentioned before as depicted in Figure 6 below. Two months' traffic prior the attack was sampled to determine the detection rate of the model.

From Figure 7 below, it is obvious, that the proposed Predictive analytics model has accurately detected a zero-day attack a few months' prior the actual attack. In October 2016, the model was already able to detect almost 60% of the traffic was prepared to the zero attack with 75% accuracy. In January 2017, 5 months before the attack, the model has detected 86% of the traffic was directed towards the attack and this time with 100% accuracy.

TABLE VIII. PROPOSED METHOD PERFORMANCE AGAINST OTHER FEATURE SELECTION METHOD

	KDDTest+_20Percent dataset		
Feature Selection methods	Detection Rate	Accuracy	False Alarm Rate (FAR)
Weighted score *w*>0	**27%** (0.272727)	**27%** (0.272727)	**72%** (0.727273)
Weighted score *w*<0	N/A	N/A	N/A
Correlation coeff. *r*>0	**11%** (0.111111)	**11%** (0.111111)	**89%** (0.888889)
Correlation coeff. *r*<0	N/A	N/A	N/A
Proposed Embedded method (Optimized)	**100%** (1.0)	**86%** (0.857143)	**14%** (0.142857)

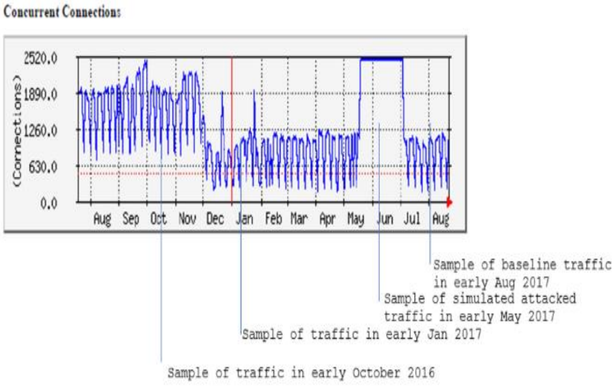

Fig. 6. Sampled Traffic.

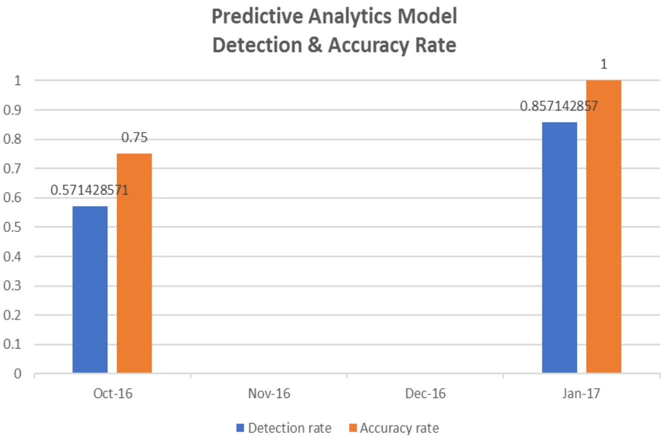

Fig. 7. Zero-day Prediction Shows the Detection Reaches 86% Detection with 100% Accuracy 5 months' Prior the Attack.

VII. Conclusion

Improve prediction and behavioural analysis. The training dataset will be trained to use the embedded feature selection method which incorporates both the filter and wrapper method. The correlation coefficient, *r* and weighted score, w_j will be incorporated. The accepted or selected features will be optimised using the Beta distribution function, *β*, to find its maximum likelihood, l_{max}. Finally, the selected features will be trained by the Bayesian Network classifier and will be tested through the inclusion of several testing datasets. Finally, this method will be compared to other feature selection methods. The results show that the proposed method's performance against other methods consistently outperforms other feature selection method. The detection rate for both NSL and KDDTest20% datasets was 100%, while KDD-Train+ scored 86%. This is because one of the tuple *Pr*(λdst_bytes) was flagged as positive which is correct. However, the intersection probability *Pr* (*BT*|λdst_bytes), or the baseline traffic given the lamda information, λdst_bytes, was actually negative. There was some reduction in the rate, otherwise it would have scored 100% as well. The False Alarm Rate was 14%, however, due to the **intersection** probability that was included in the **model**, this flag could be re-examined and re-flagged to the correct alarm. On the other hand, the detection rate and accuracy rate for the proposed optimised feature selection method scored 100% and 86%, which outperformed the other models.

Results applied onto ground-truth dataset also indicated that the prediction reaches 86% detection with 100% accuracy 5 months' prior the attack.

References

[1] M. H. M Yusof and Mokhtar M. R, "Review on Taxonomy of Malware Analysis Studies". Advanced Science Letters. 2018. Vol. 23 Issue 12.

[2] S. G. Nari. "Automated Malware Classification based on Network Behavior." 2013 International Conference on Computing, Networking and Communications, Communications and Information Security Symposium

[3] L. Xue and G. Sun. Design and Implementation of Malware Detection System based on Network Behavior. SECURITY AND COMMUNICATION NETWORKS. 2015. doi: 10.1002/sec.993

[4] R. Weaver. Visualizing and Modeling the Scanning Behavior of the Conficker Botnet in the Presence of User and Network Activity. IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. doi: 10.1109/TIFS.2015.2396478

[5] M. Zaman, T. Siddiqui, M. R Amin and M. S Hossain. Malware Detection in Android by Network Traffic Analysis. Next Generation Mobile Apps, Services and Technologies (NGMAST), 66- 71. doi: 10.1109/NGMAST.2014.57

[6] M. H. M Yusof, "A Review of Predictive Analytic Applications of Bayesian Network," International Journal on Advanced Science, Engineering and Information Technology., 2016. 6(6) ISSN: 2088-5334.

[7] G. Chandrashekar and F. Sahin. A Survey on Feature Selection Methods. Journal of Computers and Electrical Engineering 40 (2014)16-28.

[8] S.J. Stolfo ; Wei Fan ; Wenke Lee ; A. Prodromidis ; P.K. Chan.Cost-based Modeling and Evaluation for Data Mining With Application to Fraud and Intrusion Detection: Results from the JAM Project. DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings. DOI: 10.1109/DISCEX.2000.821515

[9] T. Koski and J. M Noble. Bayesian Networks. United Kingdom: John Wiley & Sons, Ltd. 2009.

[10] M. A. Ambusaidi, H. Xiangjian, N. Priyadarsi and T. Zhiyuan. Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm. IEEE Transactions on Computers (Volume: 65, Issue: 10, Oct. 1 2016).

[11] R. Kohavi and G. H. John. Wrappers for feature subset selection. Artif Intell 1997;97:273–324.

[12] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. J Mach Learn Res 2003;3:1157– 82.

[13] P. Langley. Selection of relevant features in machine learning. In: AAAI fall symp relevance; 1994.

[14] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. Artif Intell 1997;97:245–70.

[15] P. A. Mundra and J. C Rajapakse. Svm-rfe with mrmr filter for gene selection. IEEE Trans Nanobiosci 2010;9.

[16] A. Mousse. "KDD1999 dataset Features explanations". April 28, 2018. [Online].Available https://stackoverflow.com/questions/17024961/kdd1999-dataset-featuresexolaination?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

[17] A. Feizollah, N. B. Anuar, R. Salleh, F. Amalina. Comparative study of k-means and mini batch kmeans clustering algorithms in android malware detection using network traffic analysis. 2014 International Symposium on Biometrics and Security Technologies (ISBAST),2014; 193 - 197. doi: 10.1109/ISBAST.2014.7013120

[18] A. Z. Ariffin and H. Song, "Secure Knowledge and Cluster-Based Intrusion Detection Mechanism for Smart Wireless Sensor Networks," in IEEE Access, vol. 6, pp. 5688-5694, 2018. doi: 10.1109/ACCESS.2017.2770020

[19] A. Nadeem, Shukur. Z, Sulaiman. R, Current techniques in JPEG image authentication and forgery detection, Journal of Engineering and Applied Sciences.104-112. 2017.

[20] Ahmed Ali, Mohd Rosmadi Mokhtar, Loay Edwar George, 2017. Enhancing The Hiding Capacity of Audio Steganography Based On Block Mapping. Journal of Theoretical and Applied Information Technology, 1141-1148.