EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set

Muhammad Ahmad[1], Qaiser Riaz[1*], Muhammad Zeeshan[1], Hasan Tahir[1], Syed Ali Haider[2] and Muhammad Safeer Khan[3]

*Correspondence: qaiser.riaz@seecs.edu.pk
[1] Department of Computing, School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), 44000 Islamabad, Pakistan
Full list of author information is available at the end of the article

Abstract

Internet of Things (IoT) devices are well-connected; they generate and consume data which involves transmission of data back and forth among various devices. Ensuring security of the data is a critical challenge as far as IoT is concerned. Since IoT devices are inherently low-power and do not require a lot of compute power, a Network Intrusion Detection System is typically employed to detect and remove malicious packets from entering the network. In the same context, we propose feature clusters in terms of Flow, Message Queuing Telemetry Transport (MQTT) and Transmission Control Protocol (TCP) by using features in UNSW-NB15 data-set. We eliminate problems like over-fitting, curse of dimensionality and imbalance in the data-set. We apply supervised Machine Learning (ML) algorithms, i.e., Random Forest (RF), Support Vector Machine and Artificial Neural Networks on the clusters. Using RF, we, respectively, achieve 98.67% and 97.37% of accuracy in binary and multi-class classification. In clusters based techniques, we achieved 96.96%, 91.4% and 97.54% of classification accuracy by using RF on Flow & MQTT features, TCP features and top features from both clusters. Moreover, we show that the proposed feature clusters provide higher accuracy and requires lesser training time as compared to other state-of-the-art supervised ML-based approaches.

Keywords: IoT, Flow and MQTT cluster, TCP cluster, NIDS

1 Introduction

The World Economic Forum listed cyber-threat as one of the most important threats to the world economy in its 2019 Global Risk Report [1]. According to the report, companies are likely to suffer paralyzing attacks in the near term that will shut down daily operations, causing unimaginable revenue losses that exceed the breaches we have experienced to date. Such debilitating cyber-attacks would eventually lead to significant rise in investments for building adequate cyber security capabilities. Cyber-security-related expenditure is expected to reach $133 billion by year 2022, and the sector has expanded more than 30 folds in the last 13 years [2].

IoT is a collection of linked, interconnected or interlinked digital devices, mechanical equipment, entities or items, creatures or individuals, equipped with unique identification and the capacity. This involves the ability to direct information and commands over a typically wireless connection without involving interaction of humans either with computers or humans itself. With every passing day, hackers are becoming smarter and much more aggressive. According to Threatpost (a leading news site with information technology-related news), almost 98% of IoT devices traffic is in plain and more than 50% of IoT devices are vulnerable to high or medium risk [3]. With the scale, pace and complexity of today's risky environment, we ought to be capable of responding to dangers posed by such attacks in a timely and effective manner.

By detecting malicious traffic in IoT, it is ensured that IoT devices stay connected at higher level of connectivity without any interruption. Moreover, the IoT devices must also stay safe from the hackers and it can be done by keeping malicious traffic away from IoT devices. Data transmission should also be kept secure and consistent without any corruption. Efficient detection of Denial of Service (DoS) attacks is very critical in ensuring reliable communication between IoT devices. We have considered all of these problem and propose and efficient solution for detection of malicious traffic before it is transmitted through IoT device.

There are many publicly available data-sets for doing research on Intrusion Detection System (IDS). The most widely used among them are DARPA 98 [4], KDD Cup 99 [5], NSL-KDD [6] and UNSW-NB15 [7]. DARPA 98 was first made available in February 1998. It contains several weeks of network data and audit logs, but this data-set does not depict real-world traffic. KDD Cup 99 was made public in 1999 and is based on an improved version of DARPA 98. However, this data-set contains problems like duplicate and redundant records. This data-set is most widely used for IDS. NSL-KDD is a refined form of KDD 99 data-set in which the problems pertaining to KDD 99 data-set are removed. It was made public in 2009. UNSW-NB15 is the latest data-set on NIDS which came out in 2015. This data-set contains most comprehensive attack scenarios.

UNSW-NB15 data-set is created by using IXIA PerfectStorm device (used to test the security of devices) in Cyber Range Lab of Australian Centre for Cyber Security, which generated the modern real scenario-based normal and attack traffic. Besides normal traffic, it contains nine attack scenarios used in today's world. Approximately 2.5 million packets are captured and are publicly available. This data-set is available in the form of BroIDS, csv, pcap and argus. Besides full traffic, the authors also created csv files from the full data-set which contains approximately 10% of the data. This is the latest benchmark data-set for NIDS scenario.

Many researchers have used UNSW-NB15 data-set for evaluating IDS. Lopez Martin et al [8] applied a kernel approximation technique in SVM to approximate Radial Basis Function (RBF) and used combination of UNSW-NB15 data-set with NSL-KDD and Moore data-set. Moustafa et al [9] used UNSW-NB15 and Network Information Management and Security Group (NIMS) data-set to extract features relevant to Domain Name System (DNS), Hyper Text Transfer Protocol (HTTP) and MQTT attacks and applied three ML techniques of Decision Trees (DT), Naive Bayes (NB) and ANN. Zhou et al [10] used Deep Feature Embedding Learning (DFEL) technique to extract high-level features from the data-set and then uses ML techniques

(Gradient Boosting Tree (GBT), K-Nearest Neighbors (KNN), DT, Logistic Regression (LR), Gaussian Naive Bayes (GNB), SVM) using those features for evaluation. Kumar et al [11] created different number of clusters of UNSW-NB15 data-set and evaluated there efficiency using Silhouette's measure (used to check the consistency of data clusters) and then evaluated using several ML techniques based on DT.

Against the same backdrop, this work proposes a framework for defense of IoT-based networks from cyber-attacks using NIDS. The proposed model uses conventional supervised ML techniques including RF, SVM and ANN. The estimators are trained using a set of features, and the trained model is used to detect malicious traffic. Moreover, various subsets of feature set are also used in order to identify malicious traffic enabling a reduced set of feature which can be used to detect malicious traffic. We also calculate mathematical properties by combining traffic flows through possible study of flow parameters selected from MQTT and TCP parameters. After removing many discrepancies from data-set, we identify features from flow/MQTT and TCP protocols after removing the features causing over-fitting. However, top contributing features in flow/MQTT and TCP clusters are used for classification. These clusters are selected after removal of critical issues from data-set like over-fitting, imbalance nature of data-sets, the curse of dimensionality, datatype compatibility and null.

NIDS are capable of identifying malicious network traffic by using a set of characteristics that are derived from the application, transport and network layers. When positioned at a specific level or location inside a network to track data into and out of all machines on the network, the IDS can conduct a traffic forwarding analysis in order to compare the information being forwarded to a ML model which is trained on well-known and well-documented threats. Once an intrusion is detected by the IDS, warning messages can be sent to the administrator to alert them. Developing and implementing an efficient NIDS involves a data-source that includes a collection of appropriate parameters for calculating the output when categorizing normal and malicious traffic instances by using a decision-making tools. This work specifically addresses this problem by proposing efficient network detection system for IoT devices.

We structured our research in a way that we first identified the most relevant data-set related to IDS for IoT, i.e., UNSW Bot-IoT data-set. Then, the data-set is brought into the algorithm executable form by performing pre-processing. To classify the network traffic in IoT, our main focus remained on features in the data-set related to flow/MQTT protocol and TCP protocol. For comparison of our identified features in clusters, we also classify network traffic by using full features and by using only the top most contributing features in flow/MQTT and TCP protocol clusters. All four clusters are then used for classification by ML algorithms, and the accuracy results are evaluated.

Key contributions of this work are as follows:

1 Missing value imputation using three different techniques, i.e., mean, linear regression, multiple imputations.
2 Binary and multi-class classification of malicious and normal packets using full features (37) by employing three different supervised learning classifiers: RF, SVM and ANN.

3 Binary and multi-class classification of malicious and normal packets using TCP features (18) by employing three different supervised learning classifiers: RF, SVM and ANN.
4 Binary and multi-class classification of malicious and normal packets using Flow and MQTT features (13) by employing three different supervised learning classifiers: RF, SVM and ANN.
5 Binary and multi-class classification of malicious and normal packets using top contributing features selected from TCP and flow & MQTT features set (11) by employing three different supervised learning classifiers: RF, SVM and ANN.

The remainder of the paper is divided into the following sections. Section 2 covers background and previously conducted work in this field, Sect. 3 details the proposed solution, Sect. 4 explains the results and outcomes of the experiments performed using the methodology explained in Sect. 3 and compares it with the state-of-the-art and also details the analysis of results obtained in these experiments. Section 6 concludes this work.

2 Related literature

Primary focus of this work is to develop an effective NIDS that is able to detect malicious traffic in order to prevent attempts to manipulate IoT operations and resources. A considerable amount of literature had been published on NIDS [12–14]. These studies are motivated by issues arising from handling large amount of network data and dynamic nature of data.

2.1 NIDS

Industrial NIDS mainly employ either quantitative measurements or derived specifications on feature collections like packet size, inter-arrival time, stream length as well as other network data parameters to efficiently predict them in a fixed time frame [15]. They are suffering from both high false positives and false negatives rates. A significant rate of false negatives indicates that perhaps the NIDS will misidentify threats quite often, as well as a higher level of false positives implies that the NIDS will be falsely notified when there is no actual attack. Those industrial approaches are thus inadequate for threats of the modern era.

Auto-learning method is among the powerful ways of coping with attacks of the present day. These utilizes ML techniques of supervised, semi-supervised and unsupervised to identify the trends of different traditional and hostile behaviors with such a wide repository of both the Normal and threat network including happenings at the host site. While the research includes numerous solutions based on ML, the relevance toward commercial products is in initial stages [16]. The latest approaches focused on ML produce higher false positives with heavy processing costs [17]. That is because ML techniques locally learn the features of basic TCP / Internet Protocol (IP) functions.

A wide research in academics employed the analysis of the de facto standardized baseline data, KDDCup 99 to enhance the rate of Intrusion Detection (ID) effectiveness. KDDCup 99 was generated using tcpdump data from the 1998 DARPA ID assessment framework. The objective was to construct a prediction model that would divide the connection associations into Regular or Attack classes. Attacks are classified into the classes of DoS, Probe, Remote to Local (R2L), and User to Root (U2R). In the competition of KDDCup 99, as a feature structure framework, mining audit data for ID automated models (MADAMID) was used [18]. There are 41 features in MADAMID. The breakdown includes 9 packet features, 13 content features, 9 features related to traffic and 10 features are based on host. Two variants of data-set are available, complete and 10%.

The comprehensive evaluation findings of the contest of KDDCup 98 and 99 have been released in [17]. There were a total of 24 entries in the KDDCup 98 which indicated only the marked statistical importance of results in three successful entries using the variations of the DT. The 9th winner of the competition used the 1-nearest neighbor classifier. Between the 17th and 18th case, the first important output variation was noticed. This led to the initial 17 requests being rigorous and outed by [17]. The mission of the Third International Knowledge Discovery and Data Mining Tools Competition remains as a benchmark work, resulting in identification of several ML approaches. In most cases, among the results published, only 10% training and evaluation data was taken and, in few cases, the personalized data-sets were built. A detailed literature review has lately been carried out on the machine-based learning ID using KDDCup 99 data-set [19].

Following the competition, most of KDDCup 99's published findings employed many feature construction methods to reduce the dimensionality [19]. Although few researchers used custom data-sets, the bulk employed the similar data-set for freshly developed techniques in ML [19]. Such released results are in part comparable with contest results of KDDCup 99. The classification method employed in [20] comprises of P-rules and N-rules for predicting the presence and absence of class, respectively. This worked effective in competition with the preceding findings of KDDCup 99, apart from the U2R class.

For IDS with one of the most commonly deployed data-set, KDDCup 99, the importance of feature credibility assessment was explored in [21]. In [22], RF variations are addressed. They detect misuse by identifying intrusion patterns, detect anomaly by identifying mechanisms for outliers and by combining both techniques they created a hybrid technique. As related to previously published techniques in anomaly identification using unsupervised learning, the anomaly identification method using misuse approach shows improvement even from the results of top contestants of KDDCup 99 challenge. By combining the detection techniques of misuse and anomaly in hybrid system gives the advantage of improved performance [23–25].

A weak classifier of decision stumps with the AdaBoost method was employed as an ID technique [26]. With low complexity and false alarm rate and higher detection, the proposed system had better performance as compared to the results of previously published researches. However, the incremental learning was not adopted, which is a drawback. Study in [27] reported, with high detection rate the model based on shared nearest neighbor (SNN) gave the best performance. In comparison with K-means, SNN showed better performance in U2R class when experiments were performed on reduced dataset. But in their study, full data-set was not utilized.

Through networks of NB, the ID Bayesian networks were explored with root and leaf nodes representing class and features of a connection, respectively. Investigations by doing analysis through variety of experiments in ID networks using NB applications showed better performance of Bayesian networks as compared to the top competitors of KDDCup 99 challenge in Probe and U2R categories [28]. By employing Gaussian kernels with normal distribution on estimators of Parzen-window was conducted which makes this method of estimation as a nonparametric [29]. Their model, apart from intrusion data, with ensemble of DT was relatively advantageous toward the prevailing winning entries.

NIDS based on a genetic algorithm was proposed for identification of complex anomalous behavior which simplicities the modeling of spatial and temporal information [30]. In [31], we can find the overview of ID techniques based on ensemble learning, and swarm intelligence by employing the optimization technique of ant colony and its clustering and system's optimization based on particle swarm were conducted [32]. Predominant use of descriptive statistics was shown after comparing research works in related fields.

2.2 ML-based NIDS observation

To develop NIDS, numerous ML/Deep Learning (DL) techniques have been used, which includes RF, SVM, NB, Self-Organizing Maps (SOM) and ANN [12]. To reduce features, Restricted Boltzmann machine (RBM) is used and SVM is used for classification in order to implement a NIDS [33], which gives approximately 87% accuracy of the model. In combination with generative models, discriminated RBM is used for classification which achieves good accuracy [34].

For network traffic classification variants of tree-based techniques are employed, eight in total [35]. For selection of relevant features from NSL-KDD data-set, DT algorithm is used, and for classification, RF algorithm is applied. Principle component analysis (PCA) is used for selecting relevant features, and for selecting subset of optimum features, SVM is utilized [36]. A sparse encoder is developed for feature reduction in NSL-KDD data-set in conjunction with self-taught learning in order to develop a flexible NIDS [12]. They experimented their methodology on data-set with classifier of soft-max regressor and achieved an accuracy of 92.48%. [34] concluded after experimentation that the performance is decreased if the classification is performed on different training data. Work in [35] claimed after performing experiments that the accuracy is increased, and false alarm rate is decreased if the classification is performed using random tree technique.

2.3 UNSW-NB15 data-set

UNSW-NB15 is a publicly available data-set published in [7], authors of this data, gave detailed traffic-related data and also published 10% of traffic associated with the data. A larger percentage of existing literature relates to research conducted on this 10% data. For instance, in [8], authors apply ML algorithms like RF, SVM, Multi-layer Perceptrons (MLP), Convolutional Neural Network (CNN)-1D and carry out binary as well as multi-class classifications. Results show accuracy of 89.8% and 77.8% by using CNN-1D in binary and multi-class classification, respectively.

In [37], authors used this data-set to detect DoS attacks by reducing the number of features in the data. They lowered UNSW-NB15 training set to 24596 packets and test set to 68264 packets of internet traffic, including normal and DoS attack traffic. Feature set is reduced to 27 instead of 47 by removing content features, general purpose features, time fields and nominal type features in flow and basic field class. They proposed Deep Radial Intelligence with Cumulative Incarnation (DeeRaI with CuI) to detect DoS traffic. In DeeRaI, RBF with multiple abstraction levels is used to extract the intelligence. Then, the weights are optimized by using CuI and the information extracted is passed to the next level. By using DeeRaI with CuI, they achieved best accuracy of 96.15%.

UNSW-NB15 is by far the most comprehensive data-set available for testing on malicious traffic. It is for this reason that we also use this data-set for training, testing and evaluation of our proposed solution. Data parameters are given in Table 1. Each record consists of 49 features. For detailed study of features and data attributes, we encourage the reader to read [7]. We use both types of data for training and evaluation purposes.

2.4 Feature selection techniques in UNSW-NB15

Basic distribution given for UNSW-NB15 [7] is based on packets and flow features. Six categories have been defined, i.e., basic, time, content, flow, additional generated and labels. The features generated additionally are calculated from flow, basic, content and time features. It is shown that features can also be selected based on application layer protocols such as HTTP, DNS and MQTT [9]. Authors use two additional data-sets, i.e., NIMS Botnet and IoT simulation. They collect features relevant to the services from all three data-sets. ANN, NB and DT are applied along with the AdaBoost program as an ensemble process. In UNSW-NB15 data-set, accuracy achieved in DNS data records is 99.54% and in HTTP data records it is 98.97%.

Authors in [38] employ Association Rule Mining (ARM) methodology to define a relation between two or more features for selecting features with highest rank. The features are selected by comparison of UNSW-NB15 and KDD 99. The features of UNSW-NB15 are calculated in a part of KDD 99 data-set. They suggests useful attributes of most types of threats although there are certain overlapping features. Among the most repetitive attributes across all types of attacks are (1) time to live of packets from source to destination (sttl), (2) amount of rows in 100 records where srcip and dstip are same (ct_dst_src_ltm), (3) packet count from source to destination (spkts), 4) bits per second of destination (load), (5) dropped or re-transmitted packets of source (sloss), (6) dropped or re-transmitted packets of destination (dloss), (7) number of the same srcip rows in 100 rows (ct_src_ltm), and (8) rows with same dstip and service in 100 rows (ct_srv_dst).

Authors in [39] take features derived in [38] as reference and employed few techniques of feature collection from UNSW-NB15 data-set including CfsSubsetEval with GreedyStepwise approach and InfoGainAttibuteEval with Ranker procedure to determine optimal range of features. Recommended features are then extracted from

Table 1 UNSW-NB15 data-set records distribution

Type	10% data	Full data
Normal	93,000	2,218,761
Malicious	164,673	321,283
Total	257,673	2,540,044

UNSW-NB15 data-set, and RF algorithm is applied using Weka [40]. Following subset of features is implemented:

- type of service (e.g., web, ftp, smtp,... etc) (service)
- Number of bytes from source to destination (sbytes)
- time to live from source to destination (sttl)
- transmitted packet size mean by the source (smean)
- Number of rows of the same sport and dstip in 100 rows (ct_dst_sport_ltm)

In a similar work [41], authors selected ten highest ranking features from UNSW-NB15 by employing Information Gain (IG) Ranking Filter—pre-selected in [7]. Following features were selected:

- Number of bytes in a transaction from source to destination (sbytes)
- Number of bytes in a transaction from destination to source (dbytes)
- transmitted packet size mean by the source (smean)
- bits per second transmitted by source (sload)
- ct_state_ttl
- time to live from source to destination (sttl)
- time to live destination to source (dttl)
- rate (bps)
- total duration of a record (dur) (i.e., connection)
- transmitted packet size mean by the destination (dmean)

2.5 Comparison with state-of-the-art

Four classification techniques are used on these features, i.e., DT, ARM, ANN and NB for determining and discovering the roots of botnets and achieved the highest classification accuracy of 93.23% by using DT. At the cost of high computation power, the curse of dimensionality and over-fitting [8] used all 42 features given in training and test set. Out of 42, 39 features are continuous and 3 are categorical. They use a one-hot encoding on categorical data to convert it to continuous data. The resulting feature set is scaled up to 196 features. Moustafa et al [9] used UNSW-NB15, NIMS data-set, and simulated data to extract features (and created clusters) that are relevant to DNS, HTTP and MQTT attacks and applied three ML techniques of DT, NB and ANN. They analyzed the complete data-set and extracted only the packets relevant to the clusters feature set. They applied the ensemble technique and achieved the highest accuracy of 99.54% in DNS and 98.97% in HTTP.

Zhou et al [10] used DFEL technique to extract high-level features from the data-set. The fundamental idea of DFEL is to use a huge amount of data to generate high-level features and apply the model to boost the detecting speed of traditional ML algorithms. They applied ML techniques (GBT, KNN, DT, LR, GNB, SVM) using those features for evaluation and achieved the best accuracy of 93.13% using GBT in binary classification. Kumar et al [11] created a different number of clusters of UNSW-NB15 data-set and evaluated their efficiency using Silhouette's measure. The silhouette value is a measure of how similar an object is to a cluster. They evaluated those clusters using several ML techniques based on DT. Four variants of DT (C5, CHAID, CART and QUEST) have been used and achieved the best accuracy of 89.96% using the C5 variant with 22 features.

In most of the recent literature, researchers use only a small portion of UNSW-NB15 data-set, whereas in this work we use full data-set. Moreover, it is first time that we did feature selection according to network layer for UNSW-NB15 data-set. Contribution of this work includes imputation of missing values, Binary and multi-class Classification of malicious and normal packets using different combination of TCP (18), flow and MQTT features (13). We also did the binary and multi-class classification of malicious and normal packets using top contributing features selected from TCP and flow & MQTT features set (11) by employing three different supervised learning classifiers: RF, SVM and ANN.

3 Method

In this section, we discuss our proposed solution along with data-set used and ML algorithms employed for detection of malicious network traffic. Figure 1 shows an overview of steps followed in this work. At the start, we selected a data-set which is UNSW-NB15 and used it's '.csv' files. The '.csv' files had many issues like imbalance nature of data-set, datatype mismatch with the classification algorithm and also have many missing/null values. We resolved all these problems in the data pre-processing step. When the data got cleaned, the next step involved is creating different clusters according to network layers. Classification algorithms are applied to those clusters for prediction of network traffic as normal or malicious. Details of all the steps are as follows:

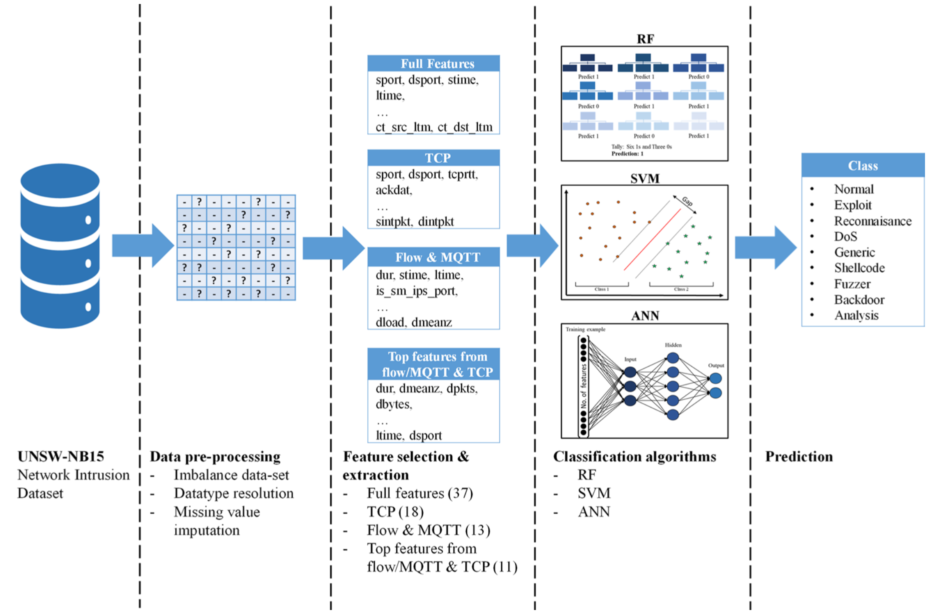

Fig. 1 Proposed framework for detecting cyber-attacks from IoT networks. Suggested system showing information flow as used in this research. UNSW-NB15 data-set is acquired and pre-processed, then features are extracted according to flow/MQTT and TCP protocols and four clusters are made. Three ML classifiers are used for classification on these clusters

3.1 Data pre-processing

Pre-processing corresponds to cleaning the data. It involves removing redundant features, features that do not render a high IG and adding derived features—features derived from other features in data. Keeping in mind that certain ML models require details in a given format, i.e., no null values allowed in RF algorithms therefore records with null values must be removed or replaced with substitute values. This issue can be resolved using imputation. Moreover, some ML algorithms cannot process data types other than integers and floats. This compatibility issue can be overcome by typecasting the values or removing the features, that do not comply, altogether. Another important dimension of pre-processing of data is that the data should be compatible with more than one algorithms for consistency and for reducing computation complexity. The pre-processing worked out on the UNSW-NB15 data-set is as follows:

3.1.1 Imbalanced data-set

Imbalance refers to an unfair class allocation within the data-set [42]. Data imbalance causes the classification to be biased. In UNSW-NB15 data-set, this problem is apparent. Class distribution percentages are shown in Table 2. Normal packets comprise of above 87% of total traffic in the data-set. We use a technique similar to undersampling of imbalanced data-set to overcome this problem. We reduced number of normal packets by 50% but kept the original number of packets of other classes. The remaining data are now 60% of the actual data.

3.1.2 Datatype resolution

Among 49 features, there are 5 features in UNSW-NB15 data-set whose data type is nominal (other than integer / float) as shown in Table 3. We removed these features from the original data-set and are left with 44 features. Second last and last features (43rd and 44th feature) are the binary and multi-class labels, respectively. The multiclass label is also of nominal type, but during algorithm execution it is converted to integer type using factorization.

Table 2 Class distribution in full data-set

Class	Percentage
Normal	87.35
Exploits	1.75
Reconnaissance	0.55
DoS	0.64
Generic	8.48
Shellcode	0.06
Fuzzers	0.95
Analysis	0.11
Backdoor	0.10
Worms	0.01

Table 3 Features with nominal data type incompatible with ML techniques

Feature	Description
srcip	IP address of the source
dstip	IP address of the destination
proto	Protocol used in the transaction
state	Shows the state of the transaction and the protocol used
service	Service used by the transaction (http, ftp, smtp, ssh, dns, ftp-data, etc)

3.1.3 Imputation of missing values

We observed that the data-set has missing values as shown in Fig. 2. Missing data can present significant bias, making the management and analysis of the data more difficult in addition to dropping the accuracy. Features containing missing values are given in Table 4. It is evident from Fig. 2 that missing values occur predominantly in three features, i.e., ct_flw_http_mthd, is_ftp_login and ct_ftp_cmd. Also, the records having most of missing values of one feature also have missing value of one or more other features. This overlapping can be seen in Fig. 2. We had two options to overcome this problem. We could remove these samples or carry out imputation. We used imputation because removing the features would negatively impact accuracy of the solution.

These substituted values, in imputation, can come from various techniques. Following are some imputation techniques which are applied on the data-set.

Table 4 Features with number of missing values

Feature	Number of missing values
sport	1
dsport	4
ct_flw_http_mthd	1,347,904
is_ftp_login	1,429,638
ct_ftp_cmd	1,429,638

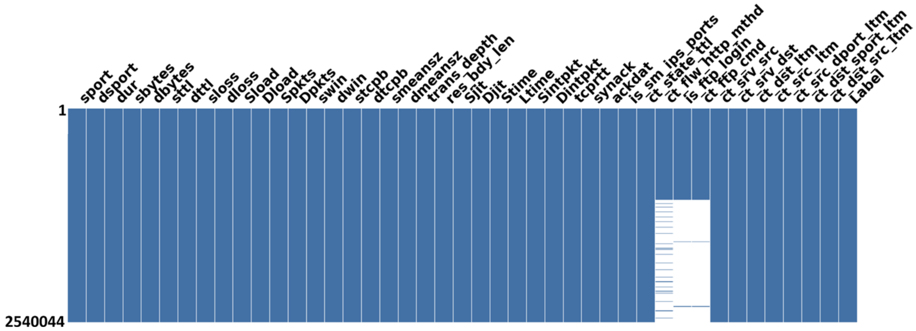

Fig. 2 Missing values matrix. Missing values in complete data-set from all features are shown in the figure. Blanks in a bar of a feature show missing values. Major portion of missing values comes from ct_flw_http_mthd, is_ftp_login and ct_ftp_cmd

1. Mean: In mean imputation, missing value on a certain sample is replaced by the mean of the values of all available samples. This method preserves the data-set size and is easy to use; however, the unevenness in the data is reduced.
2. Multiple: In multiple imputation, the missing data are filled in with expected values and a complete data-set is produced. This process of filling data-set is repeated m times, where m is the number of missing values. All of the m complete data-sets are then analyzed using a technique of concern. Imputation technique depends on the configuration of missing values as well as the type of feature(s) with missing values.
3. Linear Regression: A regression model is anticipated to guess values of a feature based on other features, and that observed model is then used to impute values in samples where the value of that feature is missing. Tailored values from the regression model are then used for imputing the missing values.

   In our solution, we apply all three imputation techniques to overcome the problem of missing values. We also compare the results achieved by applying these techniques.

3.2 Feature selection and extraction

Feature extraction is one of the core concepts in ML that has an enormous influence on prediction accuracy. The data features utilized to train ML models immensely determine our results. By using feature extraction, we reduce over-fitting, improve accuracy and also reduce training time. In our model, we use feature importance technique using RF. Five features were removed during data type resolution, and last two are labels (binary and multi-class). From the graph in Fig. 3a, it is observed that there are few features which are contributing most toward classification. This can cause over-fitting. To overcome this, we remove the top five features which has feature importance above 0.05, i.e., sbytes, sttl, Sload, smean, and ct_state_ttl. Together these five features have almost 70% of variance stored. The final feature importance graph after keeping 37 features is shown in Fig. 3b. Flow and MQTT and TCP features in Table 5. We applied imputation techniques, and the feature importance was calculated. It is observed that top features after applying the three imputation techniques separately are the same.

3.2.1 Feature clusters

In clustering, the remaining features are then clustered according to protocols. Clustering can be done according to packet and flow-based features [7] or based on a single layer services [9]. We made six clusters, i.e., flow, DNS, HTTP, File Transfer Protocol (FTP), MQTT and TCP. MQTT has been developed as a basic message protocol for equipment with restricted bandwidth. It is therefore, ideal for IoT applications. MQTT allows transmission of instructions through sensor nodes and monitors performance. Resultantly, connectivity among various devices is simple to achieve. Features in flow and MQTT are shown in Table 5. We have combined them to create a single cluster. TCP features are shown in Table 5. The third cluster of top features was extracted from flow and MQTT and TCP clusters and the features in this cluster are also shown in Table 5. Feature importance was calculated and top five and top six features from flow / MQTT and TCP were kept, respectively. The feature importance graphs of flow/MQTT, TCP and top features are shown in Fig. 4a–c, respectively.

Table 5 Shows all four clusters and the features in them

Number	Full features	Flow/MQTT features	TCP features	Top features (flow/MQTT and TCP)
1	sport	dur	sport	dur
2	dsport	Stime	dsport	dmeansz
3	dur	Ltime	dbytes	Dload
4	dbytes	Dload	sloss	Ltime
5	dttl	dmeansz	dloss	Stime
6	sloss	is_sm_ips_ports	Spkts	Dpkts
7	dloss	ct_srv_src	Dpkts	dbytes
8	Dload	ct_srv_dst	swin	Sintpkt
9	Spkts	ct_dst_ltm	dwin	Dintpkt
10	Dpkts	ct_src_ltm	stcpb	sport
11	swin	ct_src_dport_ltm	dtcpb	dsport
12	dwin	ct_dst_sport_ltm	Sjit	
13	stcpb	ct_dst_src_ltm	Djit	
14	dtcpb		Sintpkt	
15	dmeansz		Dintpkt	
16	trans_depth		tcprtt	
17	res_bdy_len		synack	
18	Sjit		ackdat	
19	Djit			
20	Stime			
21	Ltime			
22	Sintpkt			
23	Dintpkt			
24	tcprtt			
25	synack			
26	ackdat			
27	is_sm_ips_ports			
28	ct_flw_http_mthd			
29	is_ftp_login			
30	ct_ftp_cmd			
31	ct_srv_src			
32	ct_srv_dst			
33	ct_dst_ltm			
34	ct_src_ltm			
35	ct_src_dport_ltm			
36	ct_dst_sport_ltm			
37	ct_dst_src_ltm			

3.3 Classification algorithms

We use supervised learning in this work, used binary and multi-class classification on the data-set and techniques include RF, SVM and ANN. In RF, multiple DT are used to make predictions on data. Each tree predicts a class and the class predicted by most trees becomes our model prediction. SVM is characterized as a split hyper plane. It is a discriminating classifier. It provides an efficient way to classify new instances of data based on the location relative to the division (or split). The motivation for ANN is biological which comes from the nervous system structure. It is built on the lines of brain

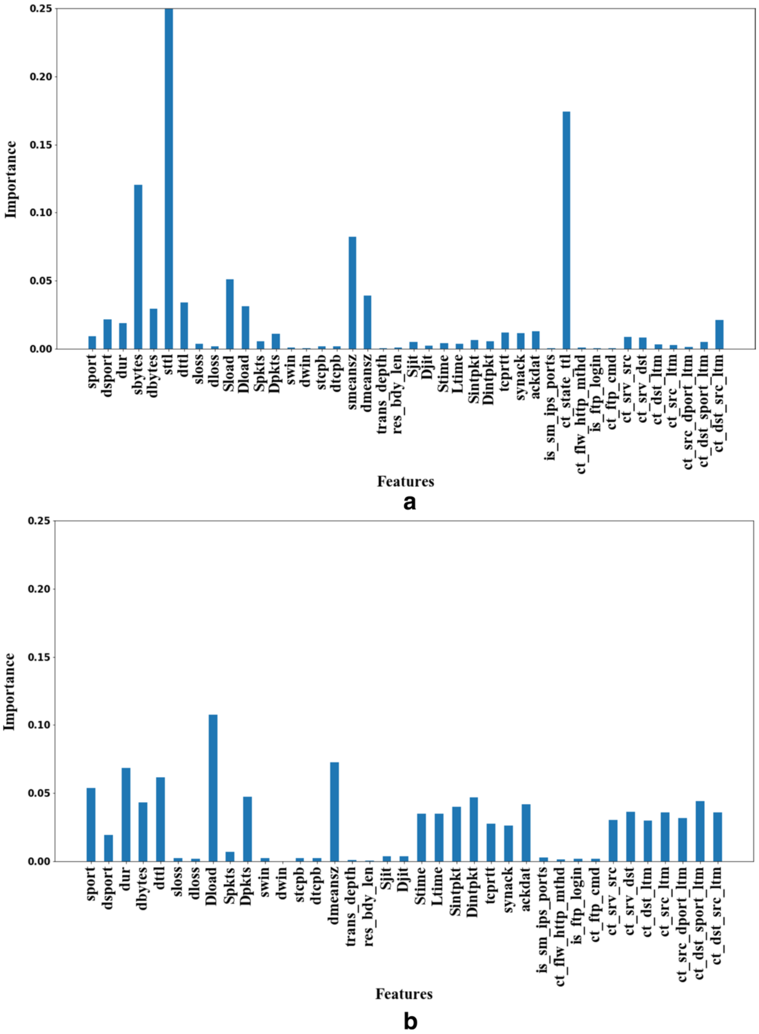

Fig. 3 Feature importance of full features. Feature importance using RF for **a** full features and **b** remaining features after removal of features having importance above 0.05. Top contributing features are, **a** sbytes, sttl, Sload, smean, and ct_state_ttl and **b** Dload, dmeansz, dur and dttl

functionality. In mammalian cerebral cortex, the way the neuronal system works is the structural basis of ANN but on a much reduced level.

3.3.1 Parameter tuning

In RF and SVM, we applied GridSearchCV on the data-set on important hyper-parameters. In RF, we applied GridSearchCV on n_estimators, criterion, max_features and min_samples_split and found out n_estimators = 100, criterion = gini, max_features = auto and min_samples_split = 5. In SVM, we found penalty = l1, loss = squared_hinge, tol = 10^{-12} and C = 10^{10} to the most optimal hyper-parameters. We use Intel (R) Xeon (R) CPU E3-1285 v6 @ 4.10 GHz with 8 CPUs, 4 cores per CPU and 2 threads per core. The system runs Ubuntu 18.04.1 LTS and has 64 GB of RAM. We also use Google Colaboratory (Colab)—free to use platform for execution of DL algorithms with pre-installed libraries. By default, it provides 12 GB of RAM and a GPU as well.

4 Results

As mentioned earlier, we performed both binary and multi-class classifications. We used reduced data-set, i.e., flow and MQTT features, TCP features and top features from flow and TCP clusters. These experiments are conducted on data with three imputations as discussed earlier. With the help of parameter tuning, the optimal parameters are computed which are used for binary and multi-class classification. We applied binary classification only on the full data-set and multi-class classification on full data as well as layer clusters.

4.1 Binary classification

In binary classification, we applied RF, SVM and ANN on optimal hyper-parameters found during parameter tuning phase. Accuracy results achieved are shown in Fig. 7a. Here, we can see that the best result in terms of accuracy score is 98.67% obtained by applying RF with mean imputation. In SVM, the best accuracy score is 97.69% also achieved with mean imputation. In ANN, the best accuracy score is 94.78% achieved with multiple imputation. Confusion matrices are shown in Fig. 5. Overall, we can see that accuracy achieved using various imputation techniques is not significantly different. The mis-classification is more significant in Fig. 5g–i, the reason is the simplicity of the model used which is verified by the confusion matrices of RF and SVM. The other reason is that the data contain too much noise in some samples, in that case a weak classifier finds it difficult to classify those samples correctly.

The confusion matrices shows the heat map of accuracy achieved by each classifier. From Fig. 5, we can see that the highest accuracy in terms of true positives and true negatives is achieved by RF followed by SVM and ANN. In ANN, we can see that percentage of mis-classification of malicious traffic with normal traffic is around 15%.

4.2 Multi-class classification

In multi-class classification, we performed experiments on reduced data-set with full features and cluster-based features and used all imputation techniques used in binary classification. All three ML algorithms are used for evaluation. Multi-class classification results using reduced data-set are shown in Fig. 7b. We achieved highest accuracy of 97.37% by applying RF on regression imputed data-set. With SVM, we achieved 95.67% accuracy, and with ANN, we achieved 91.67% accuracy. The confusion matrices for these experiments are shown in Fig. 6. Overall, from Fig. 7 we can see that RF outperformed SVM and ANN in both categories.

From the confusion matrices, we can see that the general trend of classification along the diagonal is very good except for few classes where the mis-classification accuracy is a bit higher. Mainly, we can see that the DoS packets have been mis-classified as exploits in all three algorithms. The same goes for backdoor and analysis class. This mis-classification is due to the imbalance nature of data-set. We have tried removing this issue of data-set but since the variation in available number of samples of each class is higher, therefore after even resolving this problem we still are left with resolving it for classes with fewer samples. This issue causes a certain degree of overfitting which is affecting the classification of those classes.

The comparison graphs of accuracy of all ML algorithms using feature clusters are shown in Fig. 8. We achieved highest accuracy of 96.96%, 91.4% and 97.54% in flow / MQTT, TCP and top features using RF.

4.3 Comparison with state-of-the-art

The most relevant-related research in the domain is presented in [8]. Authors have applied various ML algorithm including RF, SVM and MLP. They have also conducted binary as well as multi-class classification. In binary and multi-class classification, they achieved highest accuracy of 89.8% and 78.2% by using CNN-1D.

Authors in [9] created DNS and HTTP feature clusters and applied various classification algorithms. They considered complete data-set and extracted only the packets relevant to the cluster feature set. They applied ensemble technique and achieved highest accuracy of 99.54% in DNS and 98.97% in HTTP.

Table 6 Comparison of classification rates with existing approaches in the literature on UNSW-NB15 data-set

References	Data used	Technique	Imputation	Classification accuracy (%)				
				Binary	Multi-class	Multi-class clusters		
						Flow features	Transport features	Top features
[8]	10%	CNN-1D	–	89.80	78.20	–	–	–
		RF		87.90	73.20	–	–	–
		SVM (Linear)		84.60	65.20	-	–	–
		MLP		86.60	74.90	–	–	–
[9]	30%	DT	–	–	–	95.32 (DNS)	97.13 (HTTP)	-
		NB		–	–	91.17 (DNS)	95.91 (HTTP)	–
		ANN		–	–	92.61 (DNS)	96.27 (HTTP)	–
		Ensemble		–	–	99.54 (DNS)	98.97 (HTTP)	–
[10]	10%	GBT	–	93.13	–	–	–	–
		KNN		91.90	–	–	–	–
		DT		92.29	–	–	–	–
		LR		92.35	–	–	–	–
		NB		92.52	–	–	–	–
		SVM		92.32	–	–	–	–
[11]	10%	DT	–	89.86 (22 Features)	–	–	–	–
Proposed approach	60%	RF	Mean, multiple and linear regression	**98.67**	**97.37**	**96.96**	**91.40**	**97.54**
		SVM		97.69	95.67	89.78	82.96	89.93
		ANN		94.78	91.67	86.37	81.63	87.68

Bold values show highest accuracies achieved by proposed solution

Similarly, DFEL [10] used six different algorithms and achieved best accuracy of 93.13% using GBT in binary classification.

Four variants of DT (C5, CHAID, CART and QUEST) have been employed in [11]. Authors reduced the features set using IG and classified using varying number of features. They achieved the highest accuracy of 89.86% using 22 features with C5 variant.

It is evident from our results that our proposed solution achieved better accuracy in both classifications. The comparison is presented in Table 6.

5 Discussion

In multi-class classification on reduced data-set with full features, the three classes, i.e., DoS, backdoor and analysis, show a high false alarm rate and all three classes are mis-classified as exploits, as shown in Fig. 6. There are two main reasons behind this anomaly. First, there is a hairline difference when the feature values of these classes are compared, which makes it very difficult for ML algorithm to classify it correctly. Second, moreover, even by understanding the classical definition of all these four classes, it is very difficult for a human being to understand the difference between all these classes. However, majority of packets of these classes are classified correctly. This trend can generally be observed in all ML algorithms.

Results from Fig. 8 show that the accuracy score variation does not come from changing imputation technique but from changing feature clusters. The reason is that the most

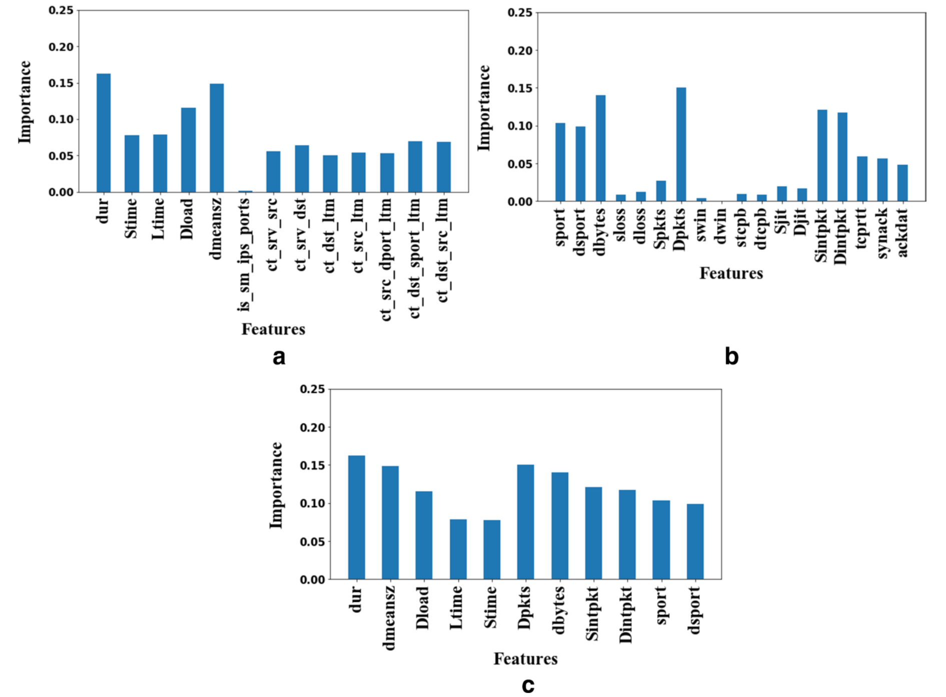

Fig. 4 Feature importance of clusters. Feature importance using RF for **a** Flow and MQTT features, **b** TCP features and **c** Top features from flow/MQTT and TCP with mean imputation. Multiple and linear regression has same trend of feature importance in all three categories. Top contributing features are, **a** sport, dttl and dbytes, **b** dloss, dur and dtcpb and **c** dur, dpkts and dmeansz. Generally, the trend toward information contribution is smooth in all three categories

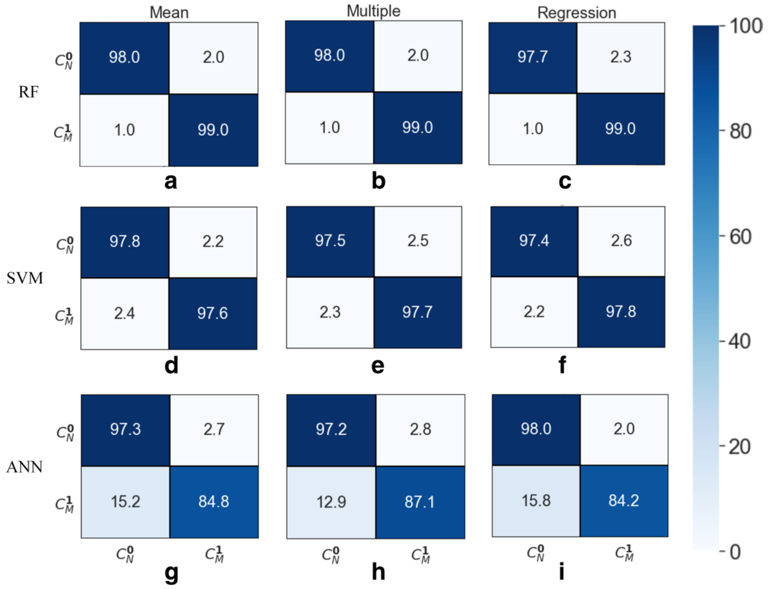

Fig. 5 Binary Classification Confusion Matrices. **a**–**c** are of RF with mean, multiple and regression imputation, **d**–**f** are of SVM with mean, multiple and regression imputation and **g**–**i** are of ANN with mean, multiple and regression imputation. In C_j^i, C is class, i is the label which is 0 or 1 and j shows that 0 is normal(N) and 1 is malicious(M) packet. In RF and SVM, there is very less % of false alarms but in ANN that % is slightly higher which is mostly due to the malicious traffic being classified as normal

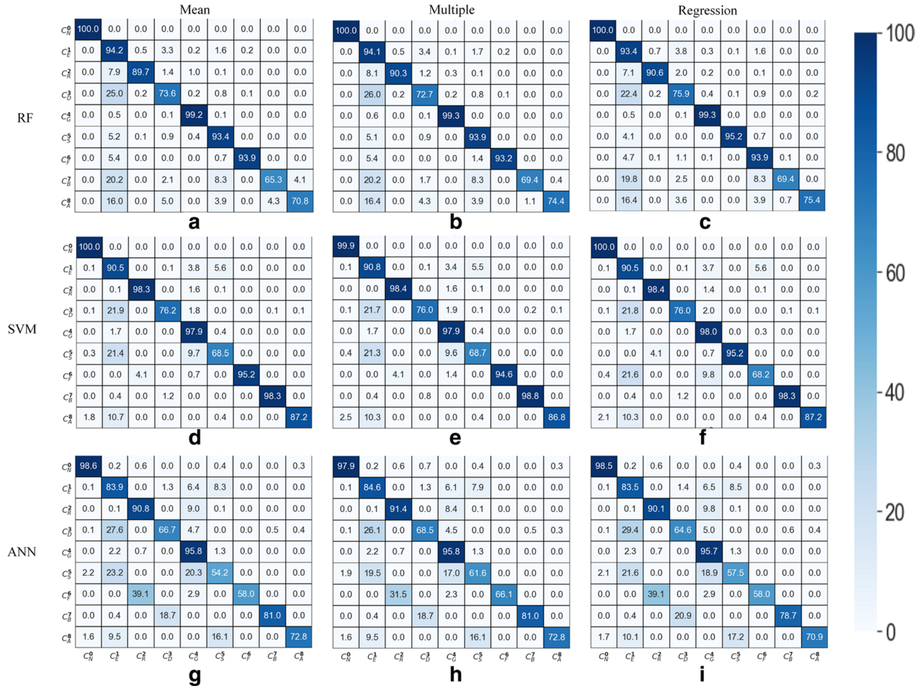

Fig. 6 Multi-class Classification Confusion Matrices. **a**–**c** are of RF with mean, multiple and regression imputation, **d**–**f** are of SVM with mean, multiple and regression imputation and **g**–**i** are of ANN with mean, multiple and regression imputation. In C_j^i, C is class, i is the label which is from 0-8 and j shows that 0 is normal and 1 is exploits(E), 2 is reconnaissance(R), 3 is DoS(D), 4 is generic(G), 5 is shellcode(S), 6 is fuzzers(F), 7 is backdoor(B) and 8 is analysis(A) packet

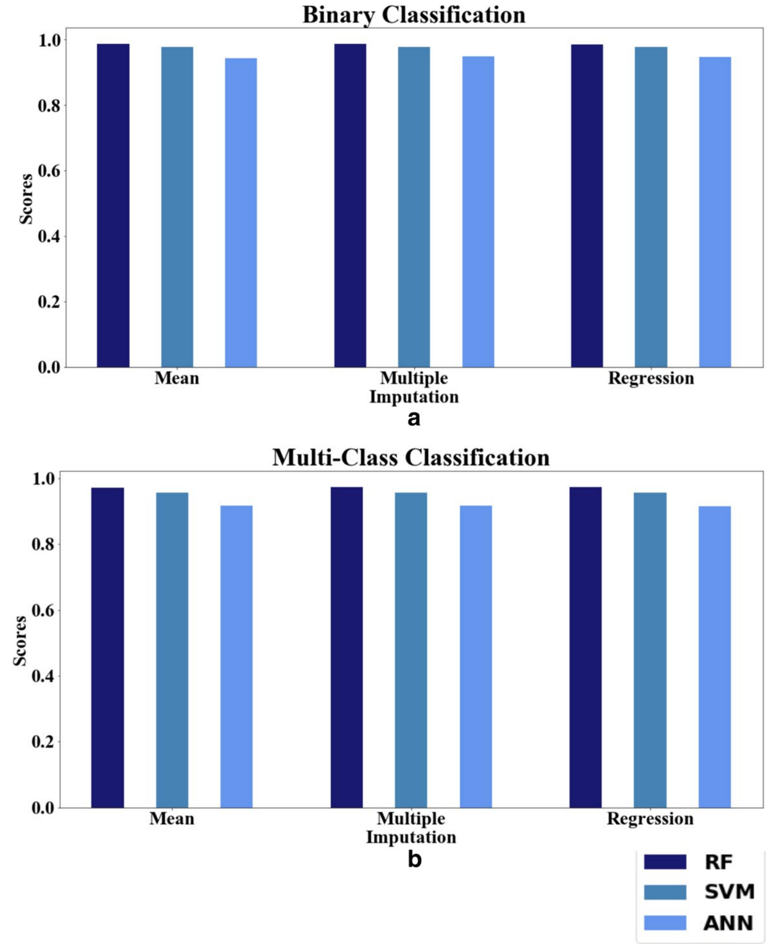

Fig. 7 Binary and multi-class classification results comparison. This figure shows the comparison of binary and multi-class classification using three imputation techniques over three ML classifiers. Generally, same trend can be seen in which RF has out-performed SVM and ANN in all imputation techniques

relevant and important features in any packet belong to flow category. When the top features from both categories are combined together, the accuracy increases. This trend can also be due to the fact that two features with missing values as shown in Table 4 requiring imputation are in TCP cluster (Table 5) and none of those features are in flow cluster (Table 5). Confusion matrices for RF are shown in Fig. 9.

In our solution, we remove the imbalance in the data-set, but to keep the training quality of algorithms we have not reduced the malicious packets instead, we have reduced the number of normal packets to remove the biased classification. Generally from Figs. 6 and 9, it can be seen that with more training examples of a class the accuracy increases. The diagonal values show promising accuracy in most cases but by using transport features only we can see that the mis-classification of three classes DoS, backdoor and analysis with exploits. This can be due to the reason that the features used for these types of attacks falls in flow category.

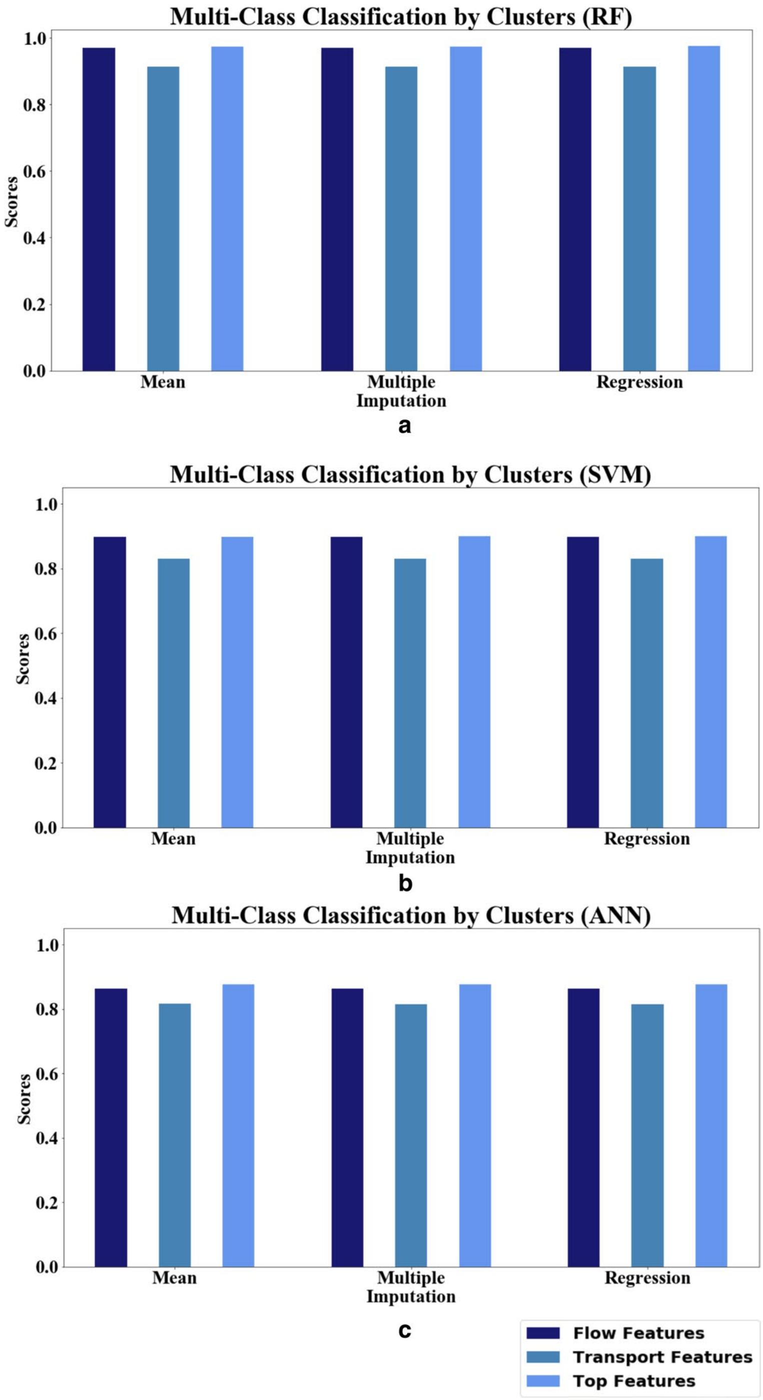

Fig. 8 Multi-class classification comparison of feature clusters In this figure, three sub-graphs of RF, SVM and ANN show classification results over three clusters. By using the top features cluster, the classification generally out-performed the other two clusters in terms of accuracy with lesser number of features

6 Conclusion

In this paper, appropriate set of features related to TCP/IP model are identified and clusters of features focusing mainly on flow, MQTT and TCP protocols are proposed.

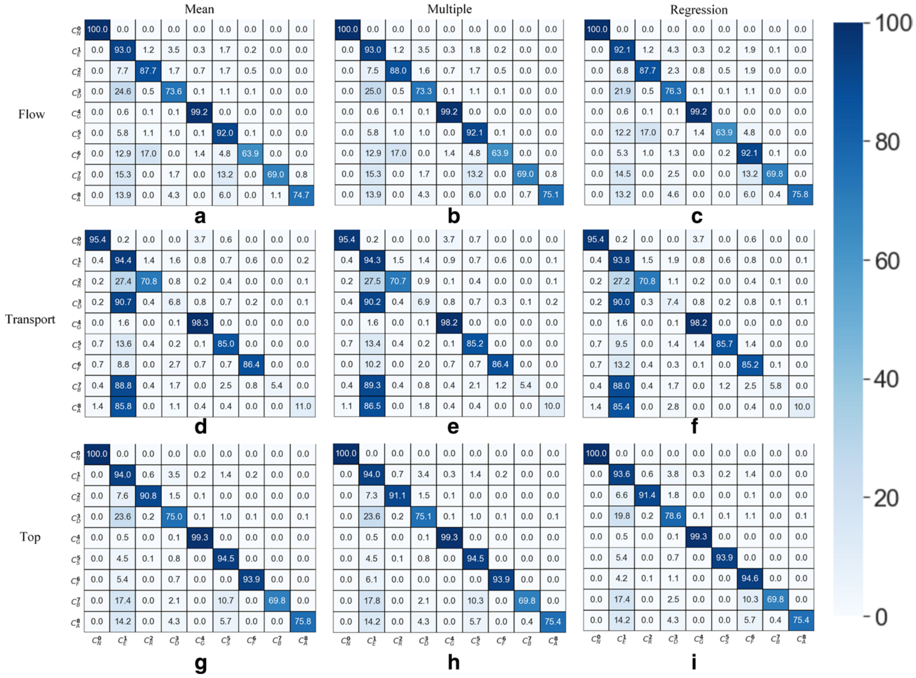

Fig. 9 Multi-class Classification Confusion Matrices on Feature Clusters Data-set using RF. **a**–**c** are of flow features using mean, multiple and regression imputation, **d**–**f** are of TCP features using mean, multiple and regression imputation and **g**–**i** are of top features from flow and TCP using mean, multiple and regression imputation

We have also removed certain issues pertaining to the full data-set which are curse of dimensionality, over-fitting and imbalanced data by removing few features and also by reducing the data-set. We used various imputation techniques to substitute missing values in data. The effects of using remaining features and using clusters are shown in terms of accuracy by evaluating the results using multiple ML algorithms. Overall, in binary classification on full data-set we achieved highest accuracy score of 98.67% with RF using mean imputation. In multi-class classification, the highest accuracy score using full data-set is 97.37% with RF using linear regression imputation. In clusters-based classification, RF seemed to outperformed other ML algorithms by achieving 96.96% in flow features, 91.4% in TCP features and 97.54% in top features from flow and TCP clusters.

In the future, in order to increase the profiling accuracy of the patterns adopted by malicious traffic, more focus will be on the collection of appropriate features related to other IoT protocols. By the use of suggested methodology and with the collection of relevant features, the detection accuracy of known and unknown attacks will increase. Moreover, other data-sets will also be analyzed using these techniques.

Abbreviations

ANN: Artificial Neural Network; ARM: Association Rule Mining; CNN: Convolutional Neural Network; CuI: Cumulative Incarnation; DT: Decision Trees; DFEL: Deep Feature Embedding Learning; DL: Deep Learning; DeeRal: Deep Radial Intelligence; DoS: Denial of Service; DNS: Domain Name System; FTP: File Transfer Protocol; GNB: Gaussian Naive Bayes; GBT: Gradient Boosting Tree; HTTP: Hyper Text Transfer Protocol; IoT: Internet of Things; IP: Internet Protocol; IG: Information Gain; ID: Intrusion Detection; IDS: Intrusion Detection System; KNN: K-Nearest Neighbors; LR: Logistic Regression; ML: Machine Learning; MQTT: Message Queuing Telemetry Transport; MADAMID: Mining Audit Data for ID Automated Models; MLP: Multi-layer Perceptrons; NB: Naive Bayes; NIDS: Network Intrusion Detection System; NIMS: Network Information Management and Security Group; PCA: Principle Component Analysis; RBF: Radial Basis Function; RF: Random Forest; R2L:

Remote to Local; RBM: Restricted Boltzmann Machine; SOM: Self-Organizing Maps; SNN: Shared Nearest Neighbor; SVM: Support Vector Machine; TCP: Transmission Control Protocol; U2R: User to Root.

Acknowledgements
Not applicable.

Authors' contributions
MAB, MZ and QR conceived the presented idea. MAB, QR and MZ developed the theory, and AB performed the computations. QR and MZ verified the analytical methods and encouraged MAB to investigate and supervised the findings of this work. MAB, QR, MZ, HT and SAH discussed the results and contributed to the final manuscript. MAB wrote the manuscript with support from QR and MZ. QR, MZ, HT, SAH and MSK reviewed the manuscript gave feedback and updated the manuscript. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The data-set analyzed in this research is UNSW-NB15 which is publicly available data-set and can be downloaded from this URL: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Department of Computing, School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), 44000 Islamabad, Pakistan. [2] State University of New York at Fredonia, New York, NY, USA.
[3] Arkansas Tech University, Russellville, AR, USA.

Received: 26 May 2020 Accepted: 8 January 2021
Published online: 21 January 2021

References

1. WEF: The Global Risks Report 2019. (2019). https://www.weforum.org/reports/the-global-risks-report-2019. Accessed Mar 2019
2. O. Yunger, Cybersecurity is a bubble, but it's not ready to burst. (2019). https://techcrunch.com/2019/10/03/cybersecurity-is-a-bubble-but-its-not-ready-to-burst/. Accessed Mar 2019
3. L. O'Donnell, More Than Half of IoT Devices Vulnerable to Severe Attacks. (2020). https://threatpost.com/half-iot-devices-vulnerable-severe-attacks/153609/. Accessed Mar 2019
4. MIT: 1998 DARPA Intrusion Detection Evaluation Dataset. Lincoln Laboratory MIT (1998). https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset. Accessed Mar 2019
5. UCI: KDD Cup 1999 Data. University of California, Irvine (1999). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed Mar 2019
6. UNB: NSL-KDD dataset. University of New Brunswick (2009). https://www.unb.ca/cic/datasets/nsl.html. Accessed Mar 2019
7. N. Moustafa, J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in *2015 Military Communications and Information Systems Conference (MilCIS)*. (Springer, 2015), pp. 1–6. https://doi.org/10.1109/MilCIS.2015.7348942
8. M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, J. Lloret, Shallow neural network with kernel approximation for prediction problems in highly demanding data networks. Expert Syst. Appl. (2019). https://doi.org/10.1016/j.eswa.2019.01.063
9. N. Moustafa, B. Turnbull, K.R. Choo, An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Internet Things J. **6**(3), 4815–4830 (2019). https://doi.org/10.1109/JIOT.2018.2871719
10. Y. Zhou, M. Han, L. Liu, J.S. He, Y. Wang, Deep learning approach for cyberattack detection, in *IEEE INFOCOM 2018—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. (Springer, 2018), pp. 262–267. https://doi.org/10.1109/INFCOMW.2018.8407032
11. V. Kumar, A. Das, D. Sinha, Statistical Analysis of the UNSW-NB15 Dataset for Intrusion Detection, pp. 279–294 (2020). https://doi.org/10.1007/978-981-13-9042-5-24
12. A. Javaid, Q. Niyaz, W. Sun, M. Alam, A deep learning approach for network intrusion detection system, in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*. (Springer, 2016), pp. 21–26
13. N. Sultana, N. Chilamkurti, W. Peng, R. Alhadad, Survey on SDN based network intrusion detection system using machine learning approaches. Peer Netw. Appl. **12**(2), 493–501 (2019)
14. B. Selvakumar, K. Muneeswaran, Firefly algorithm based feature selection for network intrusion detection. Comput. Secur. **81**, 148–155 (2019)
15. A. Azab, M. Alazab, M. Aiash, Machine learning based botnet identification traffic, in *2016 IEEE Trustcom/BigDataSE/ISPA*. (IEEE, 2016), pp. 1788–1794
16. V. Paxson, Bro: a system for detecting network intruders in real-time. Comput. Netw. **31**(23–24), 2435–2463 (1999). https://doi.org/10.1016/S1389-1286(99)00112-7

17. R.C. Staudemeyer, Applying long short-term memory recurrent neural networks to intrusion detection. South Afr. Comput. J. **56**(1), 136–154 (2015)
18. W. Lee, S.J. Stolfo, A framework for constructing features and models for intrusion detection systems. ACM Trans. Inf. Syst. Secur. (TiSSEC) **3**(4), 227–261 (2000)
19. A. Özgür, H. Erdem, A review of kdd99 dataset usage in intrusion detection and machine learning between 2010 and 2015. PeerJ Preprints **4**, 1954–1 (2016)
20. R.C. Agarwal, M.V. Joshi, Pnrule: A new framework for learning classifier models in data mining (a case-study in network intrusion detection), in *SDM* (2001)
21. H.G. Kayacik, A.N. Zincir-Heywood, M.I. Heywood, Selecting features for intrusion detection: a feature relevance analysis on KDD 99 intrusion detection datasets, in *Proceedings of the Third Annual Conference on Privacy, Security and Trust* **94**, 1722–1723 (2005)
22. J. Zhang, M. Zulkernine, A. Haque, Random-forests-based network intrusion detection systems. IEEE Trans. Syst. Man Cybernet. Part C (Appl. Rev.) **38**(5), 649–659 (2008)
23. M.S. Huda, J.H. Abawajy, M. Alazab, M. Abdollahian, M.R. Islam, J. Yearwood, Hybrids of support vector machine wrapper and filter based framework for malware detection. Future Gener. Comput. Syst. **55**, 376–390 (2016)
24. M. Alazab, S. Huda, J. Abawajy, R. Islam, J. Yearwood, S. Venkatraman, R. Broadhurst, A hybrid wrapper-filter approach for malware detection. J. Netw. **9**(11), 2878–2891 (2014)
25. T. Kim, B. Kang, M. Rho, S. Sezer, E.G. Im, A multimodal deep learning method for android malware detection using various features. IEEE Trans. Inf. Forens. Secur. **14**(3), 773–788 (2018)
26. W. Hu, W. Hu, S. Maybank, Adaboost-based algorithm for network intrusion detection. IEEE Trans. Syst. Man Cybernet. Part B (Cybernet.) **38**(2), 577–583 (2008)
27. L. Ertöz, M. Steinbach, V. Kumar, Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data, in *Proceedings of the 2003 SIAM International Conference on Data Mining*. (SIAM, 2003), pp. 47–58
28. A. Valdes, K. Skinner, Adaptive, model-based monitoring for cyber attack detection, in *International Workshop on Recent Advances in Intrusion Detection*, (Springer, 2000), pp. 80–93
29. D.-Y. Yeung, C. Chow, Parzen-window network intrusion detectors, in *Object Recognition Supported by User Interaction for Service Robots*, vol. 4, (IEEE, 2002), pp. 385–388
30. W. Li, Using genetic algorithm for network intrusion detection, in *Proceedings of the United States Department of Energy Cyber Security Group*, vol. 1, pp. 1–8 (2004)
31. L. Didaci, G. Giacinto, F. Roli, Ensemble learning for intrusion detection in computer networks, in *Workshop Machine Learning Methods Applications*, Siena, Italy (2002)
32. C. Kolias, G. Kambourakis, M. Maragoudakis, Swarm intelligence in intrusion detection: a survey. Comput. Secur. **30**(8), 625–642 (2011)
33. M.A. Salama, H. Eid, R. Ramadan, A. Darwish, A.E. Hassanien, Hybrid intelligent intrusion detection scheme. Adv. Intell. Soft Comput. **96**, 295–302 (2011). https://doi.org/10.1007/978-3-642-20505-7_26
34. U. Fiore, F. Palmieri, A. Castiglione, A. De Santis, Network anomaly detection with the restricted Boltzmann machine. Neurocomputing **122**, 13–23 (2013)
35. S. Thaseen, C.A. Kumar, An analysis of supervised tree based classifiers for intrusion detection system, in *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, (IEEE, 2013), pp. 294–299
36. L. Wang, R. Jones, Big data analytics in cyber security: network traffic and attacks. J. Comput. Inf. Syst. (2020). https://doi.org/10.1080/08874417.2019.1688731
37. N.G. Bhuvaneswari Amma, S. Selvakumar, Deep radial intelligence with cumulative incarnation approach for detecting denial of service attacks. Neurocomputing **340**, 294–308 (2019). https://doi.org/10.1016/j.neucom.2019.02.047
38. N. Moustafa, J. Slay, The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems, in *2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*. (Springer, 2015), pp. 25–31. https://doi.org/10.1109/BADGERS.2015.014
39. T. Janarthanan, S. Zargari, Feature selection in UNSW-NB15 and KDDCUP'99 datasets, in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pp. 1881–1886 (2017). https://doi.org/10.1109/ISIE.2017.8001537
40. M.L. Group, Weka 3: Machine Learning Software in Java. University of Waikato. https://www.cs.waikato.ac.nz/ml/weka/. Accessed Mar 2019
41. N. Koroniotis, N. Moustafa, E. Sitnikova, J. Slay, Towards developing network forensic mechanism for botnet activities in the IOT based on machine learning techniques, in *Mobile Networks and Management*, ed. by J. Hu, I. Khalil, Z. Tari, S. Wen (Springer, Cham, 2018), pp. 30–44
42. W. Badr, Having an Imbalanced Dataset? Here Is How You Can Fix It. Online (2019). https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb

Publisher's Note