Journal of Network and Computer Applications 150 (2020) 102499

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Real-time throughput prediction for cognitive Wi-Fi networks

Muhammad Asif Khan [a,*], Ridha Hamila [a], Nasser Ahmed Al-Emadi [a], Serkan Kiranyaz [a], Moncef Gabbouj [b]

[a] *Department of Electrical Engineering, Qatar University, P.O. Box 2713, Doha, Qatar*
[b] *Laboratory of Signal Processing, Tampere University of Technology, FIN-33101, Tampere, Finland*

ARTICLE INFO

Keywords:
Wi-Fi
Throughput
Prediction
Real-time
Machine learning
Cognitive networks

ABSTRACT

Wi-Fi as a wireless networking technology has become a widely acceptable commonplace. Over the course of time, the applications landscape of Wi-Fi networks is growing tremendously. The proliferation of new services is driving the industry to adopt novel and agile approaches to ensure the quality of experience delivered to the end user. To enhance end user experience, transmission throughput is an important metric that has a strong impact on the end-user quality of experience. The accurate real-time prediction of throughput can bring several new possibilities to enhance user experience in future self-organizing cognitive networks. However the real-time prediction of transmission throughput is challenging due to the dependency on several parameters. Previous studies on throughput prediction are primarily focused on non real-time prediction in less-dynamic networks. The studies also do not provide high accuracy as required in cognitive networks for efficient decision making. The purpose of this study is to use data-driven machine learning (ML) techniques and evaluating their accuracy and efficiency to predict the transmission throughput in Wi-Fi networks. Four algorithms are used namely multi-layer perceptrons (MLP), support vector regressors (SVR), decision trees (DT) and random forests (RF). It is widely understood that the accuracy and efficiency of machine learning (ML) algorithms hugely depend upon the datasets being used to train the model. Hence, this study proposes two distinct data models for creating ML-ready datasets using feature engineering. The accuracy of each ML algorithm over these datasets is evaluated. The evaluation results show a maximum prediction accuracy of 96.2% using MLP algorithm, followed by DT (94.5%), RF (93.3%) and SVR (91.0%) respectively. Furthermore, the complexity analysis is also presented to support the adaptation of such schemes in real-time applications.

1. Introduction

Wireless LANs today are commonplace for Internet access in homes, offices, shopping malls, metro stations, airports and other public areas. Apart from the Internet access, there are other areas where Wi-Fi networks are being used. In dense networks such as sports stadiums, conferences and exhibitions, Wi-Fi is the first choice for content distribution. Wi-Fi Direct (Alliance, 2016) based device to device (D2D) networks (Khan et al., 2017, 2019) can be deployed as an efficient tool of emergency communication and alerts dissemination. In transportation, vehicles to vehicles (V2V) and vehicles to infrastructure (V2I) communication can be enabled by deploying Wi-Fi based vehicular networks (VANETs) for sharing route information and disseminate traffic alerts. In e-commerce, the D2D Wi-Fi networks (i.e. Wi-Fi Direct) can be utilized to send targeted proximity based advertisements. In Internet-of-Things (IoT), Wi-Fi networks are considered as a potential candidate for gateway nodes to collect real-time sensor data and forward to the central controllers.

The aforementioned services and applications open new challenges to Wi-Fi networks in terms of Quality of Service (QoS) and Quality of Experience (QoE) required by the end user. Among several others, the transmission throughput is a significant metric to measure the user experience. For instance, multimedia-based applications have high throughput requirement as compared to other applications such as web-browsing and email transfer. Thus, a high number of users running multimedia applications can cause higher packet delays, jitters and congestion in the network, which leave a negative impact on the overall user experience. In the next-generation cognitive networks, the net-

[*] Corresponding author.
E-mail addresses: mkhan@qu.edu.qa (M.A. Khan), hamila@qu.edu.qa (R. Hamila), alemadin@qu.edu.qa (N.A. Al-Emadi), mkiranyaz@qu.edu.qa (S. Kiranyaz), moncef.gabbouj@tut.fi (M. Gabbouj).

https://doi.org/10.1016/j.jnca.2019.102499
Received 4 July 2019; Received in revised form 2 October 2019; Accepted 18 November 2019
Available online 22 November 2019

works shall be equipped to cope with such impairments by predicting network changes ahead of time. For instance, the accurate throughput prediction in the event of a new device requesting to associate with a congested access point (AP) can help the network to decide whether the new device shall be offered connection or not? Similarly, knowledge of throughput over different transmission channels in advance, can be used to select the best channel in overlapping basic service set (BSS) to mitigate inter-BSS interference. Furthermore, such prediction models can bring significant advantages to design efficient network topologies as well as optimum scheduling mechanisms (Bui et al., 2014; Sundaresan et al., 2015; Samba et al., 2016).

Despite knowing the benefits of throughput prediction, the accurate prediction of transmission throughput in Wi-Fi networks can be challenging due to the dependency on several parameters such as, the number of users in the network, the quality of the wireless connection (i.e. Signal-to-Noise Ratio or SNR), the capabilities of access point (AP) and client devices (i.e. support for latest 802.11 standards, MIMO support), the data transmission channel and the channel bandwidth (usually 20 or 40 MHz in 802.11n). The prediction accuracy becomes more challenging in highly dynamic and dense networks due to the user's mobility, interference between user's devices in same BSS, interference with adjacent BSS, hidden and exposed terminals, frequent device's association/de-association and the diversity of user applications (Kurose and Ross, 2013; Zahmatkesh and Kunz, 2017).

Analytical models based on these parameters to predict the transmission throughput might not be efficient due to several reasons. Firstly, the wireless channel is always changing which makes the estimation of connection quality difficult. Secondly, users are usually mobile in most Wi-Fi deployments. The mobility causes random channel variations which impact throughput. Thirdly, in highly dynamic networks, connected devices can frequently disassociate from the networks and new devices are likely to connect to the network, thus the network density is also variable. Fourthly, the capabilities of these devices is another challenge e.g. the existence of 802.11 g, 802.11n and 802.11ac devices in the networks causes different rates for the same quality of channel. Lastly, different devices connected to the network run different applications and might have different traffic generation rates and patterns, which has a direct impact on throughput. These diverse conditions make the prediction of QoS metrics such as transmission throughput challenging and mathematical models suffer from inaccuracies to predict the exact transmission throughput. Furthermore, any mathematical model which provides even the closest approximations is always based on assumptions which can not be fully justified in the real world. Thus, instead of relying on mathematical approximations which have several limitations, machine learning algorithms can be employed to predict the transmission throughput. The advantage of ML based techniques is their capability to learn and adapt to the varying environment and network conditions. Such ML based prediction schemes can be used in future self-organizing cognitive networks to significantly improve user experience.

In this article, we propose in this article to employ ML techniques to predict the transmission throughput in Wi-Fi networks. The ML-based techniques typically rely on historical data with useful features in appropriate structure and format suitable to train the ML algorithm. By feeding the datasets to the ML algorithm, the algorithm learns the behavior of the particular network to predict the desired output. To build a robust and accurate ML model, the choice of learning algorithm can play important role in prediction performance and accuracy. We carefully reviewed several ML algorithms and limit our selection to four i.e. compact feed-forward Artificial Neural Networks (ANNs), or the so-called Multi-Layer Perceptrons (MLPs), Support Vector Regressor (SVR), Decision Trees (DT) and Random Forest (RF). There are several reasons for choosing these techniques: Firstly, the throughput prediction is a regression problem solved using supervised learning, hence the learning algorithms that are more suitable for regression problems are selected. The four algorithms are also used by other works which involves problems of similar nature (Kousias et al., 2019; Chen and Huang, 2018; Samba et al., 2016; Mirza et al., 2010). Secondly, simpler models such as Multiple Linear Regression (MLR) are not used because of the complexity of the prediction problem. The transmission throughput is a non-linear and non-stationary function and linear models like MLR do not offer sufficient prediction accuracy. Thirdly, deep learning algorithms such as Convolution Neural Networks (CNN), Long-Short Term Recurrent Neural Networks (LSTM-RNN) offers stronger generalization capability, however these are avoided due to longer training time and large data samples requirements. Fourthly, deep learning algorithms are best suited when the available datasets are very large. In contrast, when the available dataset is small, simple ML algorithms may outperform deep learning as also shown in (Zhang et al., 2019).

The contributions of this paper are:

- After a thorough search in publicly available repositories of network datasets, the datasets with the required features are not found. Hence, we collected raw network data using simulations of Wi-Fi network in Mininet-Wifi SDN (Software-Defined-Networks) emulator. To further validate the significance of the study, network traces are collected from real Wi-Fi network using our Ubuntu-based packet sniffer with Atheros Wi-Fi chipset and a 9-dBi omnidirectional antenna.
- A detailed analysis of several network attributes collected from network traces and their impact on the transmission throughput is performed to aid in feature engineering process.
- The network traces in raw form are not suitable for use in the proposed prediction problem. Hence, new features are extracted from the raw data. Two distinct data models are proposed to produce ML-ready datasets. The datasets are published on the public repository to serve as benchmark datasets, which are freely available at (Khan, 2019).
- Highly accurate prediction of transmission throughput is achieved in real-time which can be employed in future cognitive networks for several applications.

The rest of the paper is organized as follows: Section 2 presents a review of the state-of-the-art in machine learning applications in the field of communication networks with focus on predicting QoS metrics. Section 3 explains the various stages of the proposed scheme i.e., the acquisition of raw network traces, selection of the network attributes, extraction of a set of useful features for prediction, preprocessing, and a brief description of the machine learning algorithms used for prediction. The actual implementation details are presented in Section 4 which particularly focuses on investigating the impact of each individual feature on the target variable, selection of a subset of important features and tuning of hyperparameters. Section 5 evaluates the performance of the ML algorithms in the proposed scheme over the benchmark datasets created for prediction accuracy, speed, and robustness. Conclusions are drawn in Section 7 along with some useful research directions to extend and to further improve the proposed approach.

2. Related work

Recently, several studies have been carried out to demonstrate the significance of machine learning in wireless communication in several applications (Zhang et al., 2019; Anwar et al., 2019; Jiang et al., 2017; Kato et al., 2017; Fadlullah et al., 2017; Assra et al., 2016; Wen et al., 2015; Feng and Chang, 2012; Xia et al., 2012; Choi and Hossain, 2013; Donohoo et al., 2014). Supervised learning algorithms such as regression models, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM) or Support Vector Regressor (SVR) can be applied in channel estimation, user localization (Feng and Chang, 2012) and energy learning (Donohoo et al., 2014). Similarly, Bayesian learning can be applied in multiple-input, multiple-output (MIMO) for channel learning and spectrum sensing using Gaussian Mixture Models (GMM) (Wen et al., 2015), Expectation Maximization (EM) (Choi and Hossain, 2013) and Hidden Markov Model (HMM) (Assra et al., 2016). Unsupervised learning algorithms such as K-means clustering can be used to build optimal topologies in Device-to-Device (D2D) networks for energy efficiency and overall network efficiency (Xia et al., 2012). Principal Component Analysis (PCA) and Independent Component Analysis (ICA) can be efficiently utilized in applications of anomaly detection, fault isolation, and intrusion detection. Multi-Layer Perceptrons (MLPs) as a sub-class of Artificial Neural Networks have been applied to several problems in next-generation wired and wireless networks in several applications (Kato et al., 2017) (Fadlullah et al., 2017).

Authors in (Zhang et al., 2019) propose an auto-learning framework for wireless networks using machine-learning. The study explains how machine learning helps to solve complex network optimization problems where existing optimization techniques might fail.

The accurate prediction of future traffic load can play significant role in predictive resource allocation. Authors in (Guo et al., 2018) proposed a predictive model using SVR and Long Short Term Memory (LSTM) algorithms to predict the average bandwidth in cellular network. The study shows that SVR outperform LSTM. In (Chen and Huang, 2018) authors investigated the impact and potential benefits of prediction in improving timely throughput. Authors in (Mirza et al., 2010), addressed the problem of throughput prediction for TCP flows in Wide Area Networks (WAN). The authors highlighted issues related to traditional methods of predicting TCP throughput which uses time series forecasting and presented their approach which is based on Support Vector Regression (SVR). The dataset used to predict TCP throughput is obtained using a laboratory testbed. The prediction accuracy is evaluated using “relative prediction error” metric proposed in (He et al., 2005).

Authors in (Kousias et al., 2019) used supervised machine learning algorithm to estimate downlink throughput in mobile broadband networks. The authors used 39 parameters related to mobile networks to create 14 features. Several models are built, trained and tested using multiplier linear regression (MLR), SVR and RF algorithms. The results show that RF outperform SVR and MLR algorithms in terms of accuracy. The study reports highest complexity of RF algorithm followed by SVR and MLR respectively.

In (Wei et al., 2016), authors used Hidden Markov Model (HMM) algorithm to predict future throughput values of next 100 s in mobile networks. The authors compared the performance of their proposed model against traditional history-based forecasting methods including linear regression and stochastic model.

In (Liu and Lee, 2015), authors addressed the prediction of TCP throughput in cellular (3G/HSPA) networks. The authors used seven prediction algorithms and compared the prediction accuracy of each using the root-mean-squared error (RMSE) metric. The prediction model uses past throughput samples to predict the future values. The TCP throughput prediction used in (Liu and Lee, 2015) is approximately a linear functions which is simpler than the transmission throughput as addressed in this paper.

In (Samba et al., 2016), authors proposed a throughput prediction strategy in Long Term Evolution (LTE) cellular network using several network parameters such as RSSI, Signal-to-Noise Ratio (SNR), Reference Signal Received Quality (RSRQ) and Reference Signal Received Power (RSRP). The authors used three machine-learning algorithms namely Generalized Linear Model (GLM), Artificial Neural Networks (ANN) and Random Forests (RF), to evaluate the predictor performance. The study shows that random forest outperforms GLM and ANN by showing better generalization.

Authors in (Yue et al., 2017) proposed a similar model to predict throughput in cellular network. However, the model uses parameters available in cellular networks (i.e. RSRP, RSRQ, CQI, handover events and past throughput values). The model uses random forest algorithm to predict the next 1sec throughput.

3. Proposed scheme

We propose in this paper to employ ML based techniques to accurately predict the transmission throughput in Wi-Fi networks in varying environmental and network conditions. Our proposed model is different from all the aforementioned works discussed in Section 2. For instance, none of these works has addressed the prediction of transmission throughput in Wi-Fi networks. Most of the referred works are focused on TCP throughput (Mirza et al., 2010; Bui et al., 2014; Sundaresan et al., 2015; Liu and Lee, 2015). The prediction of TCP throughput is a different problem which becomes much simpler in wired networks. On the other side, the overall network throughput is impacted by all TCP and UDP sessions running in network devices. Furthermore, the throughput prediction in Wi-Fi networks is relatively more complex function as explained in Section 1.

The work in (Samba et al., 2016) addresses the transmission throughput in cellular networks. The proposed work in this paper is different from (Samba et al., 2016) because, in (Samba et al., 2016) the authors aim to predict the transmission throughput for a file download application, whereas the proposed work considers several applications such as video streaming, video calls and web browsing simultaneously. Secondly, authors in (Samba et al., 2016) predict the throughput once for the duration of file download, which is a relatively simpler as compared to prediction over short time window as considered in the proposed scheme. Furthermore, the cellular network being larger in network density, the impact of variations for few stations has less impact on the overall throughput, thus making the prediction less sensitive as compared to the prediction in Wi-Fi networks. Moreover, the authors also did not provide details about the features used in this model and the way to acquire or create features, whereas we presented a complete framework which provides all details about the data acquisitions, datasets creation, pre-processing, ML model formation, tuning and evaluation.

A flowchart of the proposed scheme is shown in Fig. 1. The steps involved in the proposed scheme are briefly described in the subsequent sub-sections.

3.1. Data acquisition

The first step in this study is the formation of the benchmark dataset for training and evaluation (test) of the predictor model. As discussed in Section 2, no prior study exists on the real-time prediction of transmission throughput in Wireless LANs. Also, the dataset needed to train the ML algorithms in the proposed scheme are also not available.

The required datasets shall essentially contain all features that can affect the transmission throughput. These features are computed from various network attributes i.e. the number of stations connected to the access points, signal strength at each station, modulation scheme, data rates, inter-arrival time, packet arrival rate, number of retransmissions and channel parameters. All such attributes are obtained from network traces collected using tools such as Wireshark (Combs et al., 2008) or (Jacobson VanCraig and McCanne, 1989). To implement the proposed scheme, we created benchmark datasets in two different ways.

3.1.1. Synthetic dataset by simulations

Simulated data reduces the time and effort for data acquisition and allows to control several network configuration parameters which usually are hard to control in a real environment. To perform simulations of Wi-Fi networks, we use Mininet-Wifi (Fontes et al., 2015) which is an SDN (Software Defined Networks) emulator and extends the functionalities of Mininet (M. Team, 2012) to emulate Wi-Fi networks. We chose Mininet-Wifi due to several reasons: Firstly, it can create virtual Wi-Fi interfaces using the 802.11 SoftMAC wireless LAN driver, which allows us to emulate the Wi-Fi protocol control messages passing between virtual wireless access points and virtual mobile stations. The

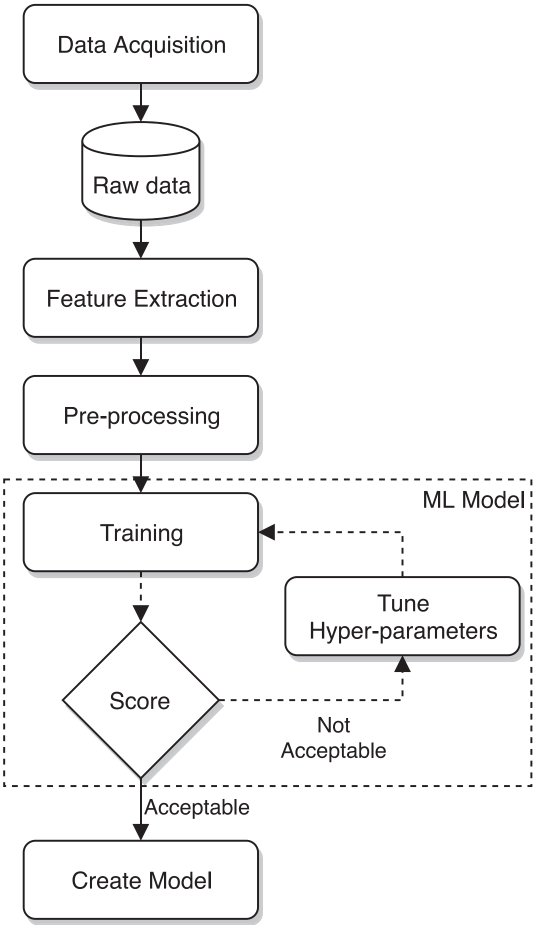


Fig. 1. Machine learning model.

control messages over Wi-Fi interfaces are essentially required for our dataset. Secondly, Mininet-Wifi allows to monitor the wireless traffic using practical tools such as Wireshark (Combs et al., 2008) or tcpdump (Jacobson VanCraig and McCanne, 1989). Finally, Mininet-Wifi unlike other network simulators such as NS-3 (Network Simulator-3), allows interactive simulation and user can add traffic and applications on devices as well as apply some topological changes during the simulation runtime. With such features, more dynamic scenarios can thus be created.

3.1.2. *Real dataset by network sniffing*

A network sniffer comprised of a set of hardware and software tools which can monitor and capture all wireless traffic in a real network. The minimum hardware requirement is a Linux machine equipped with a Wi-Fi adapter that supports “monitor” mode. In monitor mode, the Wi-Fi adapter can listen to all traffic transmitted on a particular channel. Software tools such as Wireshark or tcpdump are then used to capture all packets transmitted inside the network.

3.2. *Feature extraction*

The synthetic and real data acquired through simulations and network sniffing contain per-packet information such as time-stamp, source address, destination address, packet length, RSSI, packet type, MCS index, and data rate. The raw data is further is further used to compute some other network attributes, such as Packet Arrival Rate (PAR), Inter-Arrival Time (IAT). These additional attributes are usually calculated over time intervals, rather than per packet basis. For instance, we define a time window of fixed duration (1 s) and compute new features over this defined window as follow:

- *Number of Transmitting Stations*: Wireless medium is shared between the access point and the client stations. As the number of devices increases, the share of each device to access the channel in a given interval decreases which reduce the average throughput of the network. The increasing density of the devices transmitting simultaneously can cause collisions and thus results in packets loss which further reduces throughput.
- *Number of Receiving Stations*: We are interested in the average throughput, hence the number of receiving stations is directly affecting the transmission throughput.
- *Number of Active Clients*: Although the network size is a significant parameter to predict interference, a more significant parameter which affects the throughput is the number of devices in active state or more importantly, in transmit or receive mode. A node in sleep state or idle mode usually has no or a minimal impact on the throughput.
- *Received Signal Strength Indicator*: The RSSI values provide a direct indication of the perceived quality of the wireless link. A device with higher RSSI values usually transmit/receive packets with a higher data rate and thus have higher transmission throughput.
- *Packet Arrival Rate*: The packet arrival rate (PAR) at a station indicates the throughput demand of the application. However, if evaluated over a small fixed duration, the PAR also indicates the link quality and the adapted rates.
- *Packet Transmission Rate*: The packet transmission rate (PTR) at a single station indicates the application’s packet generation rate, the adapted rates, and the channel access rates of the station.
- *MCS Index*: The modulation and coding scheme (MCS) index is selected based on the SNR parameters. Different 802.11 standards correspond to different MCS selection matrices. The MCS index is a direct mapping of transmission rates and thus it indicate the network throughput.
- *Supported Rates*: The client devices in a BSS may not have equally capabilities. There may be legacy devices which supports only 802.11 b and/or 802.11 g rates while some devices will be equipped with 802.11n and even 802.11ac rates.
- *Channel*: If there are other Wi-Fi networks in the common transmission range, the selection of non-overlapping channel help to avoid interference, reduces collisions and improve throughput. The channel can be automatically selected by AP or manually assigned. If the channel selection is set on *auto* mode in the AP configuration, this parameter will be an important attribute.
- *Number of Retransmissions*: The number of retransmissions can be easily retrieved from raw traces by analyzing the sequence number and “Retry” flag in the frame header. Retransmissions can adversely affect the transmission throughput.

The extracted features from the raw data need to be represented in an appropriate format to improve the machine learning performance. Boolean inputs such as Guard Interval which takes two values = [short, long] will be encoded as a single feature, where 1 represent *long* and −1 represents *short*. Similarly, if a node is active during an observation window should be labelled as 1, otherwise labelled as − 1. Categorical inputs which may take a value from a finite set can be encoded in several ways. One approach is to use “effects” coding, similar to 1-of-(C-1) coding. For instance, the 802.11 standard supported by Wi-Fi clients is represented using effects coding as 802.11a = (−1, −1), 802.11 b = (−1, 1), 802.11 g = (1, −1) and 802.11n = (1,1). Effect coding increases the number of features in the training data. A better approach is to encode categorical inputs by their relative proportions. Thus, we can represent n-valued variable using (n-1) features without missing any information.

3.3. Pre-processing of data

ANNs may not work well without pre-processing the input data properly. Pre-processing the input data involves *(i) scaling*, *(ii) redundancy elimination by dimension reduction* and *(iii) outliers removal*. Feature scaling on a single scale usually helps the machine learning model to better characterize the distinct patterns in the dataset. Similarly, redundant or highly correlated features also have a negative effect on the model performance and hence less significant features should be removed. Outliers are data points which are either insignificant or distinct from the rest of the data points in the dataset. Outliers removal is of key importance in prediction because the model when trying to accommodate the outliers also deteriorates the learning performance on other crucial data points.

Standardization in statistics is referred to as the process of transforming a variable by subtracting the mean and dividing by the standard deviation to standard normal (zero mean, unity standard deviation). Standardization has the advantage that it increases the performance of the optimization algorithm. However, it should be treated with caution as it discards certain information contained in original values and might have an adverse effect in some cases.

3.4. Machine learning algorithms

3.4.1. Multi-layer perceptrons (MLPs)

Feed-forward and fully-connected ANNs (or MLPs) are supervised learning algorithms used to model the complex non-linear relationship between one or more inputs (features) and a real-valued output (target). MLPs are layered architectures typically arranged as one input layer, one or more hidden layers and one output layer. All the extracted features are fed to the input layer. In most typical implementations, the size (i.e. number of neurons) of the input layer is equal to the number of features. The size of the output layer depends on the application, more precisely on the type of output. The number of hidden layers and the size of each hidden layer are both considered as hyperparameters and shall be manually tuned to some practical values. Other parameters include the activation function, learning rate and optimization algorithm.

3.4.2. Support vector regressor (SVR)

Support Vector Machines (SVMs) have been extensively applied in the data mining and machine learning communities for the last decade in various domains. SVMs are typically used for learning classification, however, they can also be used for regression. The performance of SVR (SVM used for regression) algorithm relies on the selection of appropriate kernel functions to construct the model. The commonly used kernel functions are, Linear, Polynomial, Sigmoid and Radial Basis Function (RBF). The kernel function transforms the dataset from non-linear space to linear space. The kernel trick thus allows the SVR algorithm to find a fit or hyper-plane to map the data to the original space. Other hyperparameters include kernel coefficient and penalty parameter.

3.4.3. Decision tree (DT)

A Decision tree used for the regression problem is also called Regression Tree. A decision tree is comprised of a root node, decision nodes and leaf nodes. A decision node has two or more branches, each representing values for the attribute. The topmost decision node in a tree which corresponds to the best predictor called root node. Leaf node represents a decision on the numerical target. A Decision tree is a non-parametric algorithm and can model arbitrarily complex relations between inputs and outputs, without any apriori assumption. Decision trees are good to handle heterogeneous data (ordered variables, categorical variables, or a mix of both). Decision trees intrinsically implement feature selection, making them robust to irrelevant or noisy variables (at least to some extent). The significant hyper-parameter that can improve the performance of the DT algorithm is the maximum depth of the tree. Further details about the decision tree algorithms can be found in (Timofeev, 2004).

3.4.4. Random forest (RF)

Random Forest (RF) (Liaw Wieneret al., 2002) is a combination of several decision trees. The hyperparameters for RF algorithm include the number of trees, maximum features in an individual tree and minimum number of leaf nodes that are required to split an internal node.

3.5. Evaluation metrics

We evaluate the performance of our model using three regression metrics, namely (i) *Mean Absolute Error (MAE)*, (ii) *Mean Squared Error (MSE)* and (iii) *R-Squared*:

3.5.1. Mean Absolute Error

MAE measures the average of the magnitude of the errors on the test set with m data points.

$$MAE = \frac{1}{2m}\sum_{1}^{m}|y_i - \hat{y}_i| \tag{1}$$

where, y_i is the actual value of the target variable, $\hat{y}_i$ is the predicted value of the target variable, and m represents the total number of data points in the test set. The MAE given in equation (1) measures the magnitude of the error, but does not provide the direction.

3.5.2. Mean Squared Error

An alternate measure of the error performance is Mean Squared Error (MSE) which is the average of squared errors and calculated using equation (2).

$$MSE = \frac{1}{2m}\sum_{1}^{m}(y_i - \hat{y}_i)^2 \tag{2}$$

Both MAE and MSE are negatively-oriented scores, which means lower values are better. MAE is preferred over MSE when the behaviour of the outliers does not matter.

3.5.3. R-squared

Another useful metric to measure the performance of the regression model is *R-Squared* which is also known as the coefficient of determination. The *R-squared* metric provides an indication of how well the model predicts unseen values. *R-squared* is calculated using equation (3).

$$r2 = 1 - \frac{SSE}{SSTO} \tag{3}$$

where *SSE* is the sum of squared error, and *SSTO* is the total sum of squared values.

$$SSE = \sum_{i=1}^{m}(\hat{y}_i - \bar{y})^2$$

$$SSTO = \sum_{i=1}^{m}(y_i - \bar{y})^2$$

The value of *R-squared* varies between 0 and 1. A value close to 1 is always desired which indicates a good fit of predicted values to the actual values.

4. Implementation

As stated earlier in Section 3, two methods are used to acquire the raw data. In the first method, synthetic data is produced using network simulations performed in Mininet-Wifi. We simulated a Wi-Fi network consisting of 10 Wi-Fi stations and a single access point (AP). The parameters for the AP and stations are given in Table 1.

Table 1
Simulation settings.

Parameters	Access Point	Stations
Transmit Power (dBm)	14	14
Antenna Gain (dB)	5	5
Channel	1	1
Range (m)	34	18
Mode	802.11 g	802.11 g
Datarate (Mbps)	54	54
Frequency (Ghz)	2.412	2.412

In addition, we collected real Wi-Fi network traces from a real Wi-Fi network via network sniffing. In this approach, we deployed a network sniffer to capture all packets transmitted in the Wi-Fi network. We used a Linux machine along with external Wi-Fi dongle having Atheros chipset fully capable of monitor mode, along with a 9 dBi omnidirectional antenna. The Linux machine is configured in monitor mode and packets transmitted only in the home network were captured using tcpdump. The Wi-Fi stations connected to the AP includes three iPhone handsets, two Android devices, and one laptop. The access point is configured without any security features, on channel 1 in 2.4 GHz mode using 802.11 b/g/n wireless modes. All stations are 802.11n enabled.

4.1. Creation of datasets

The raw traces collected from simulations and real Wi-Fi networks as mentioned above are transformed into useful features using two different data models.

- **Model I:** In this model, we computed per-station statistics form the attributes mentioned in Table 2. The features included in this model are number of transmitting stations, number of receiving stations, mean RSSI, mode of MCS index, mean data rate, mean inter-arrival time, mean packet arrival rate and number of retransmissions. Thus, we computed total 38 features for the real Wi-Fi network (6 stations) and 62 features for the simulated network (10 stations). In this model, the number of features increases with an increasing number of stations.
- **Model II:** In this model, we computed network-wide attributes and their statistics. The features included in this model are number of transmitting stations, number of receiving stations, RSSI (min, max, mean and standard deviation), MCS index (min, max, mode, mean, standard deviation, skewness and kurtosis), data rate (min, max, mode, mean, standard deviation, skewness and kurtosis), proportion of frames sent with 40 MHz bandwidth, proportion of frames sent with long guard interval, inter-arrival time (min, max, mean, standard deviation, skewness and kurtosis), packet arrival rate (min, max, mean, standard deviation, skewness, and kurtosis) and number of retransmissions. We computed 22 features for simulated Wi-Fi network and 33 features for real Wi-Fi networks using this model. The difference in the number of features in both model is due to the unavailability of some attributes in the simulated dataset such as physical 802.11 standard, channel bandwidth, guard interval, and MCS Index. Therefore, the feature vector dimension in this model remains fixed regardless of the number of stations in the network.

The training data produced by either model is represented as a multidimensional array $(X^{(i)}_{(j)}, y^{(i)})$, where $X \in R^{(n_x, m)}$ and $y \in R^{(1,m)}$.

$$X = \begin{bmatrix} X^{(1)}_{(1)} & X^{(2)}_{(1)} & X^{(3)}_{(1)} & \cdots & X^{(m)}_{(1)} \\ X^{(1)}_{(2)} & X^{(2)}_{(2)} & X^{(3)}_{(2)} & \cdots & X^{(m)}_{(2)} \\ \vdots & & & & \\ X^{(1)}_{(n)} & X^{(2)}_{(n)} & X^{(3)}_{(n)} & \cdots & X^{(m)}_{(n)} \end{bmatrix}$$

Each row of $X \in R^{(n_x, m)}$ represents a single feature and its length (i.e., m) represents the number of data points in the dataset. The length of the column vector in $X \in R^{(n_x, m)}$ (i.e., n) represents the total number of features in the dataset. Similarly the target variable $y \in R^{(1,m)}$ is represented as a one-dimensional array of length m.

$$y = \begin{bmatrix} y^{(1)}_{(1)} & y^{(2)}_{(1)} & y^{(3)}_{(1)} & \cdots & y^{(m)}_{(1)} \end{bmatrix}$$

we used the python based highly optimized *scikit-learn* framework (Pedregosa et al., 2011) to implement the proposed scheme. This framework offers several sophisticated functions for data analysis, preprocessing and machine learning algorithms and evaluation.

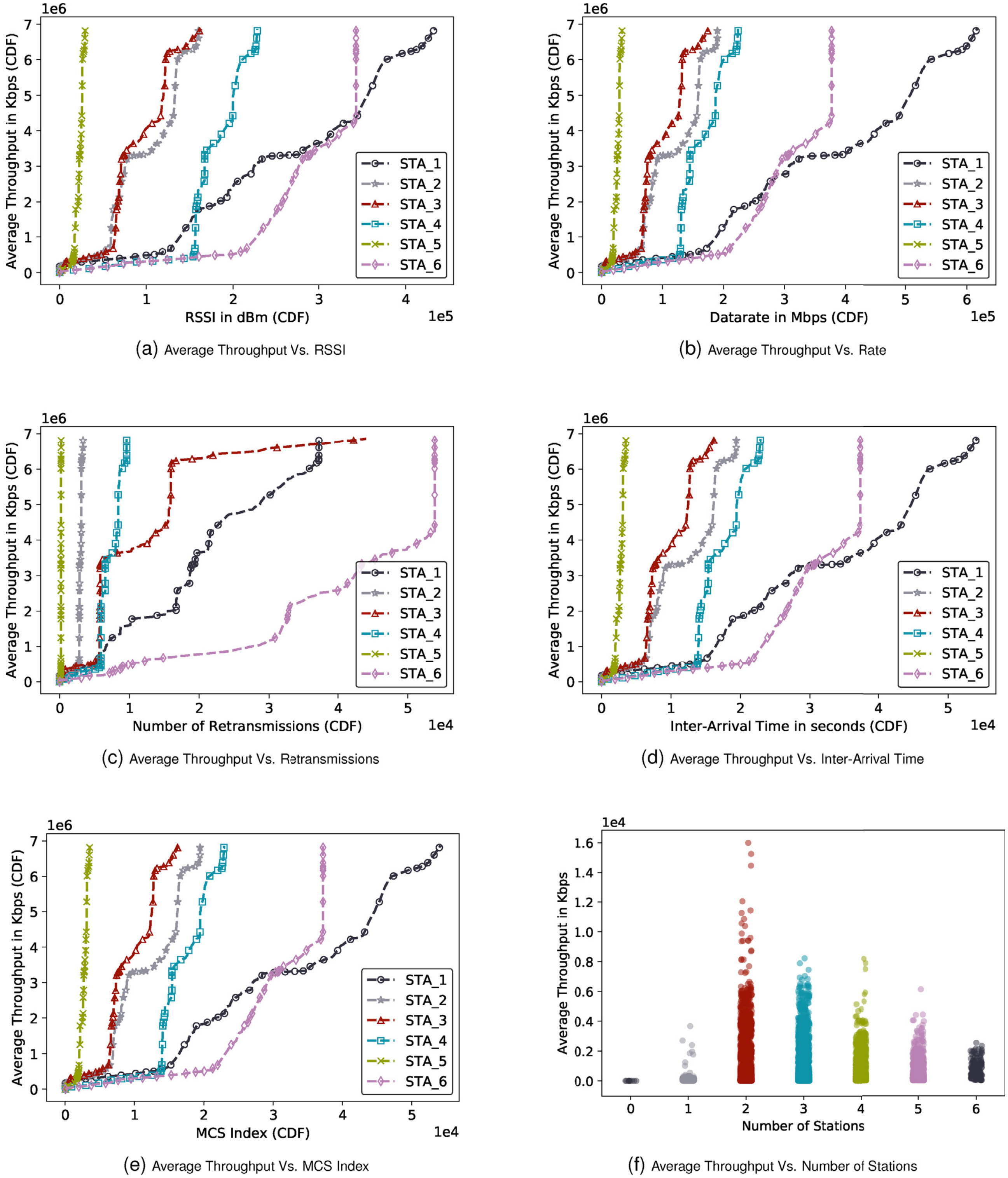
4.2. Analysis of datasets

In the first place, we analyzed our datasets to understand the relationships between different features with the target variable. Some of these features such as the number of transmitting stations, number of receive stations, MCS index are categorical variables while others such as RSSI, inter-arrival time etc. are continuous variables. A graphical illustration of these relationships for the dataset (Khan, 2019) collected from the real Wi-Fi home network is shown in Fig. 2. Both the horizontal and vertical axes show cumulative values of the labelled variables.

A careful inspection of Fig. 2 reveals the direct relationship between individual features and the target variable. The analysis shows that the impact of any single input variable on the target variable (i.e. *average throughput*) is hard to exactly describe for the acquired dataset, because the target variable simultaneously depends on other variables. Therefore, the relationship of any input variable to the target variable is complicated and requires understanding of the dataset and domain knowledge. For instance, in Fig. 2a, the dependency of *average throughput* versus *RSSI* of individual stations is shown. As expected, the usual trend in the graph is that *average throughput* increases when *RSSI* of the

Table 2
Important parameters and feature extraction.

Packet Fields	Network Attributes	Features	
		Model-1	Model-2
Source Address	Number of transmitting stations	Actual value	Actual value
Destination Address	Number of receiving stations		
Signal Strength	RSSI of received packet	mean value	minimum, maximum, mean, mode, skewness, kurtosis
MCS Index	MCS Index	mode	minimum, maximum, mean, mode, skewness, kurtosis
Rate	Data rate	mode	minimum, maximum, mean, mode, skewness, kurtosis
Time Delta	Inter Arrival Time (IAT)	mean	minimum, maximum, mean, mode, skewness, kurtosis
Arrival Time			
Type Subtype	Packet Arrival Rate (PAR)	actual value	actual value
Guard Interval (GI)	Guard Interval (short/long)	proportion of frames with long GI	proportion of frames with long GI
Bandwidth (BW)	Channel Bandwidth (20/40 MHz)	proportion of frames with 40 MHz BW	proportion of frames with 40 MHz BW
Type “Retry” Flag	Number of retransmissions	actual value	actual value
Length Type	Throughput	actual value	actual value



(a) Average Throughput Vs. RSSI

(b) Average Throughput Vs. Rate

(c) Average Throughput Vs. Retransmissions

(d) Average Throughput Vs. Inter-Arrival Time

(e) Average Throughput Vs. MCS Index

(f) Average Throughput Vs. Number of Stations

Fig. 2. Impact of parameters on throughput.

station increases. However, in some points (e.g. for Station-1), the *average throughput* tends to increase despite no increase in the *RSSI* occurs, which might be due to changes in other variables e.g. increase in the *RSSI* value of other stations. We can estimate at this point that the *RSSI* value of Station-1 have no or a minimal impact on the *average throughput* and usually such feature should be removed from the feature set to improve the speed and performance of the predictor. A similar relationship can be seen in Fig. 2b. In Fig. 2c, the impact of *retransmissions* versus the *average throughput* is presented. It can be noticed that the number of retransmissions for Station 1, 3 and 6 does not increase, which might be due to the fact that these stations are running UDP (User Datagram Protocol) and hence no retransmissions are recorded for these stations. For the rest of the stations, the rate of growth of *average throughput* is negatively affected by increasing number of retransmissions in the network. The relationship of *average throughput* versus *inter-arrival time* is illustrated in Fig. 2d. It can be observed that the *inter-arrival time*

Table 3
Hyperparameters values for ML algorithms.

Hyper-Parameters	Simulation		Real Wi-Fi Network	
	Model I	Model II	Model I	Model II
MLP				
No. of units in first hidden layer	42	71	54	33
Initial learning rate	0.01	0.031	0.001	0.01
Activation function	Relu	Relu	Relu	Relu
Optimization algorithm	Adam	Adam	Adam	Adam
SVR				
kernel	RBF	RBF	RBF	RBF
kernel coefficient	0.0002	0.085	0.01	0.01
penalty parameter	18	50	1	1
RF				
maximum depth of the tree	39	42	20	20
number of trees	10	10	10	10
DT				
maximum depth of the tree	10	28	20	20

has always an inverse relationship with the *average throughput*. At some points, the *inter-arrival time* does not increase, which means that the station is in idle mode, however, the increase in *average throughput* is due to the fact that the other stations are in active mode.

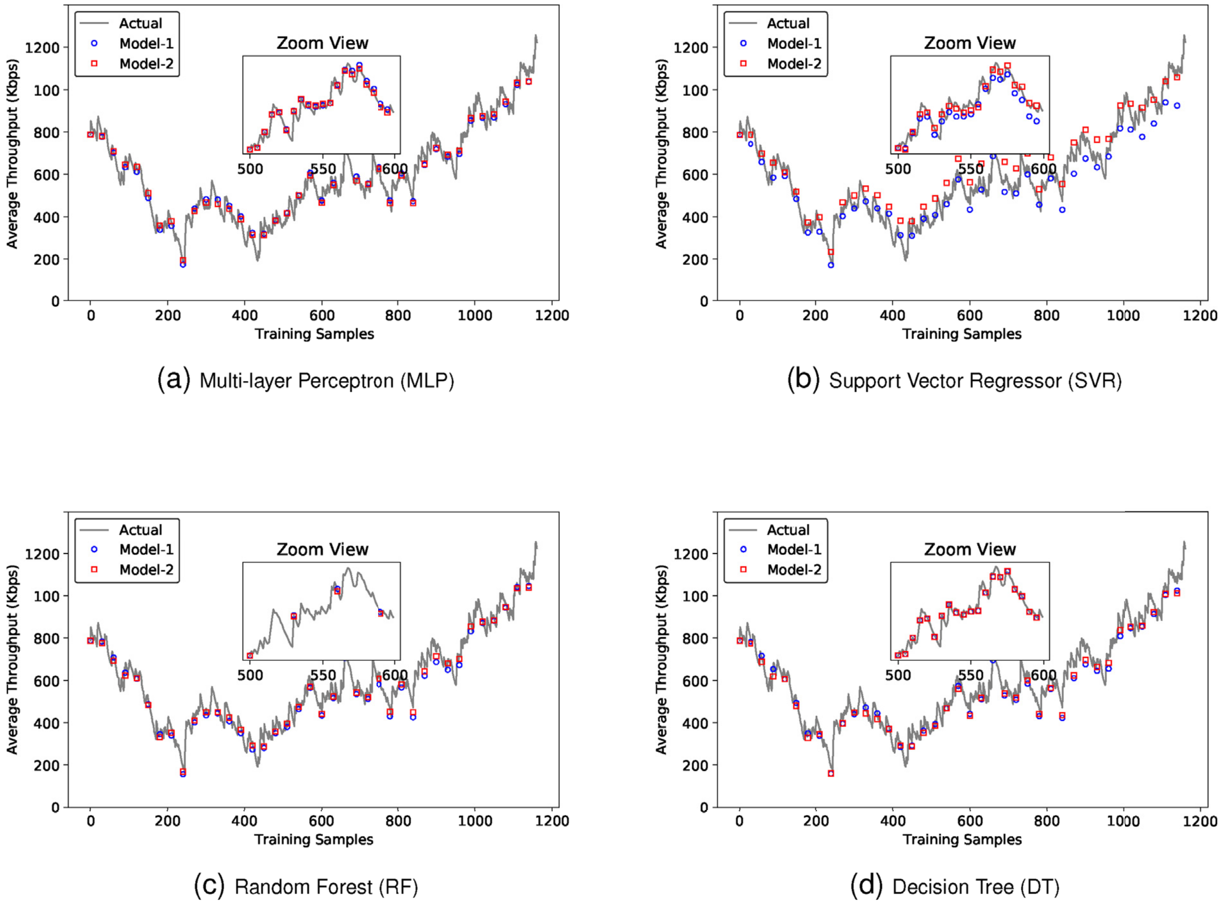
4.3. Feature selection

We created benchmark datasets which include a wide selection of features, but not all of them are equally important or some of them might be redundant since they might correlate with other features or the target variable in the dataset. Moreover, some features are noisy and may even have a negative impact on the performance in terms of both accuracy and time. It is always a good practice in large datasets with many features, to analyze all features and select a subset of features without compromising the desired accuracy of the model. In feature selection, the domain knowledge is the primary tool to decide whether a feature should be dropped or kept in the training data. For instance, Fig. 2f and e shows the impact of number of *active stations* and the *MCS index* on the *average throughput* respectively. Fig. 2b and e also show that the *Data rate* and *MCS index* has similar relationship with *average throughput*, and hence one of these features can be dropped. To further improve the model performance, we used "data normalization" technique to normalize the data as explained in Section 3.3. PCA (Andrzej and Ratajczak, 1993) can be used to reduce the least important information of the used features.

4.4. Hyperparameters tuning

The next step after feature selection is the formation of the predictor model. To train the machine learning predictor model, the dataset is first split into train (X_{train}, y_{train}) and test sets (X_{test}, y_{test}). The model is trained over the entire train set and then it is evaluated over X_{test}. The performance of machine learning algorithms highly depends upon the selection of optimum values of hyperparameters.

The hyperparameters for MLP are *number of hidden layers*, *number of neurons at each hidden layer*, *learning rate*. Apart from hyperparameters, a proper selection of other parameters such as activation function, training algorithm and regularization parameters can further improve the prediction performance.



(a) Multi-layer Perceptron (MLP)

(b) Support Vector Regressor (SVR)

(c) Random Forest (RF)

(d) Decision Tree (DT)

Fig. 3. Predicted versus actual throughput (synthetic data).

The hyperparameters for SVR include *penalty parameter of the error term*, *kernel coefficient* and the *kernal function*.

The hyperparameters for Random Forest (RF) and Decision Tree (DT) include *maximum depth of the tree*, *number of trees* (in RF) and *number of features* (in DT).

Two methods are usually recommended for hyperparameters tuning; (i) grid-search and (ii) random search. The trade-off between the two methods is accuracy vs. efficiency. We used the grid-search method to find the best possible values of these parameters and hyperparameters. The setting values of hyperparameters for each algorithm are listed in Table 3. These values are then used to build the prediction model.

5. Results and analysis

In this section, we present the performance of the two models and their effectiveness to accurately predict the *average throughput* when applied to the test dataset (X_{test}).

5.1. Prediction performance

The performance of each ML algorithm on both data models (i.e., Model-I and Model-II discussed in Section 4) is evaluated on both simulated and real network datasets. To validate the statistical significance of the acquired results, we performed 10-fold cross-validation for each ML algorithm to validate the acquired results. The cross-validation is an important step to avoid over-fitting in the model.

The predicted values of *average throughput* using each ML algorithm, versus the actual values are plotted for synthetic and real network datasets in Figs. 3 and 4. The plots show the prediction accuracy of each ML algorithm over both models.

The results listed in Tables 4 and 5 show that both data models can be efficiently used for prediction. As depicted in Tables 4 and 5, Model-I provides better prediction accuracy (i.e. less MSE values) than Model-II in most cases because the actual network attributes are used as features in Model-I, whereas Model-II is comprised of derived features using statistics of original network attributes.

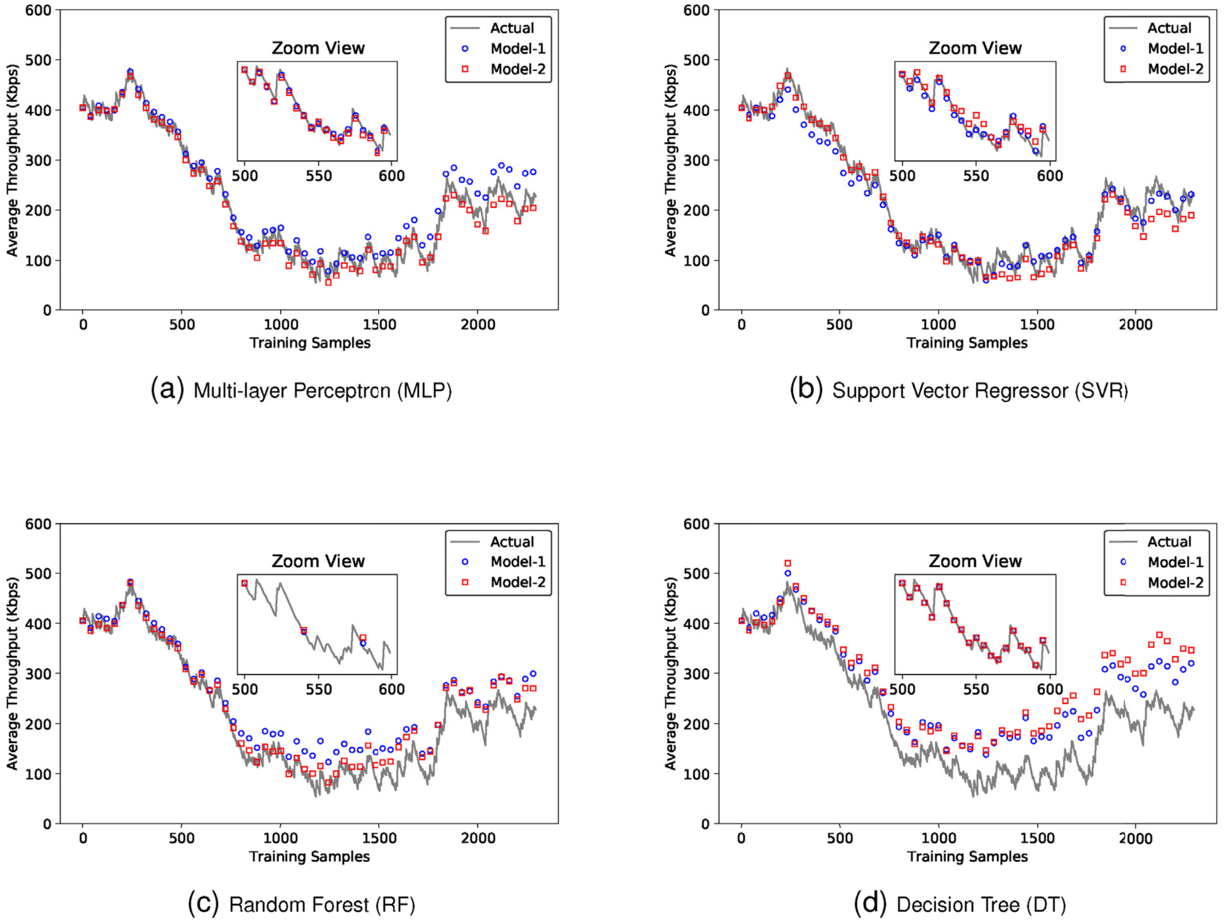
The acquired results for real datasets (Table 5) are more significant and recommended to investigate the effectiveness of the two data models as well as the performance of the ML algorithms.

The comparison between the four ML algorithms using real datasets show that MLP offers highest accuracy over both models (i.e. Model-I = 96.2%, Model-II = 94.4%), followed by DT (94.5%, 94.3%), RF (93.3%, 92.5%) and SVR (91.0%, 86%) respectively. Furthermore, the MLP and RF algorithms shows the highest generalization (99%) for both data models, followed by DT. The SVR algorithm showed the least generalization (96% and 97%) for Model-I and Model-II respectively.

5.2. Model complexity analysis

The performance evaluation, in terms of prediction capability, of the two models (i.e., Model-1 and Model-2) in (Section 3.2) is presented in 5.1. To further evaluate the significance of each algorithm using the two data models, we evaluate their complexity in this section.

We used an HP laptop equipped with Intel Core-i5 processor, 8 GB memory, Scikit-learn version 0.19.1 and Python 3.6 to implement and evaluate the proposed scheme. Using the default mode of the scikit-



(a) Multi-layer Perceptron (MLP)

(b) Support Vector Regressor (SVR)

(c) Random Forest (RF)

(d) Decision Tree (DT)

Fig. 4. Predicted versus actual throughput (real data).

Table 4
Performance Evaluation (synthetic data).

Metric	MLP		SVR		RF		DT	
	Model-I	Model-II	Model-I	Model-II	Model-I	Model-II	Model-I	Model-II
MSE	0.026	0.019	0.106	0.114	0.040	0.016	0.056	0.021
MAE	0.003	0.001	0.035	0.028	0.005	0.001	0.012	0.002
R Squared	0.99	0.99	0.96	0.97	0.99	0.99	0.98	0.99

Table 5
Performance Evaluation (real data).

Metric	MLP		SVR		RF		DT	
	Model-I	Model-II	Model-I	Model-II	Model-I	Model-II	Model-I	Model-II
MSE	0.038	0.056	0.090	0.142	0.067	0.075	0.055	0.057
MAE	0.007	0.011	0.033	0.059	0.036	0.069	0.023	0.033
R Squared	0.99	0.98	0.96	0.93	0.96	0.92	0.97	0.96

Table 6
Model complexity analysis.

Task	Synthetic Dataset		Real Dataset	
	Model-I	Model-II	Model-I	Model-II
MLP				
Feature extraction time(s)	0.0101	0.0014	0.0138	0.0029
Total Training time (s)	4.1498	1.0021	23.9228	29.0508
Prediction time (s)	0.0007	0.0003	0.0015	0.0009
SVR				
Feature extraction time(s)	0.0101	0.0014	0.0138	0.0029
Total Training time (s)	0.2864	0.1663	0.8163	0.8469
Prediction time (s)	0.0383	0.0142	0.0952	0.1078
RF				
Feature extraction time(s)	0.0101	0.0014	0.0138	0.0029
Total Training time (s)	1.1141	0.5806	1.1915	1.7753
Prediction time (s)	0.0042	0.0039	0.0075	0.0066
DT				
Feature extraction time(s)	0.0101	0.0014	0.0138	0.0029
Total Training time (s)	0.1108	0.1361	0.1646	0.3424
Prediction time (s)	0.0004	0.0005	0.0008	0.0010

learn, one processor core is used to run ML algorithms. The complexity of each algorithm is evaluated by computing the time elapsed by each model at different stages.

Table 6 presents the complexity analysis of each algorithm over both models. It can be observed that the time taken by Model-II to extract a single data point of the training set is higher than Model-I by 0.0037 s (0.3x times) for the synthetic dataset and 0.0015 (2.0x times) for real dataset. The longer time required by Model-II was expected as it involves extraction of more complex features such as mean, standard deviation, mode, skew and kurtosis from a network attribute. Moreover, the time difference is less in the case of synthetic dataset because of less number of features in Model-II for the synthetic dataset. Thus it can be concluded that Model-II requires more computational time than Model-I in feature extraction stage. The feature extraction time is independent of the ML algorithms.

In the training stage, the training time (i.e. the time required to train the ML algorithm) is highest for MLP followed by RF, SVR and DT. The DT algorithm outperforms the three algorithms with respect to training time using both data models. The benefit is validated using both synthetic and real datasets. In the prediction stage, the prediction time is highest for SVR, followed by RF using both data models. The least prediction time is offered by MLP over Model-II and DT over Model-I. The relationships are validated over both synthetic and real datasets.

The training and prediction time over synthetic dataset for Model-I is found less than that of Model-II. To the best of our understanding, the reason for this contrasting behaviour over synthetic dataset is the lack of some network attributes in synthetic dataset caused by the limitations of the simulation software which resulted in longer training and prediction time for Model-II.

The aforementioned complexity analysis provides information in terms of the time required to train the model and the prediction of future values. Another aspect of the proposed scheme is the implementation of the proposed ML based models in real networks. The proposed scheme requires several network-related parameters to create useful features. These parameters are readily available in packets information. However, the acquisition of these parameters from each packet and then extraction of features from these parameters may not be practical in current network deployments. However, in future cognitive networks based on Software Defined Networks (SDN) architecture, the acquisition and processing of these features is fairly possible. In these architectures, the computationally expensive data processing tasks are performed by the *SDN Controller*. The *SDN controller* is equipped with enough memory and processing capability to implement complex processing functions. As compared to data acquisition and processing, the model training is performed periodically and not continuously. Once the model is trained, it is stored in the memory and the stored model is run when prediction is required. The prediction time is relatively negligible.

6. Performance comparison

In Section 5, the performance of the proposed scheme is validated by considering four machine learning algorithms on two distinct data models. In this section we further compare performance of the proposed scheme against other legacy methods for throughput prediction as well as similar methods used previously for throughput prediction in the literature.

Table 7
Comparison with other methods.

Ref	Scheme	MSE	MAE
Kim et al. (2007)	ARIMA	0.316	0.285
Liu and Lee (2015)	EWMA	0.347	0.316
Liu and Lee (2015)	NN	0.231	0.201
Liu and Lee (2015)	SVR	0.277	0.246
Proposed	MLP (Model-I)	0.038	0.007
Proposed	MLP (Model-II)	0.056	0.011

One of the most commonly used statistical methods used for forecasting a non-stationary time-varying function (also called as time series) is Auto Regressive Integrated Moving Average (ARIMA). It is considered as a simple yet powerful method in time series forecasting and has been used in wireless communication in several prediction problems (Kim et al., 2007). Due to its powerful prediction capability, we applied the algorithm to the throughput dataset. We tried several values of the *lag* parameter to tune the performance and chose the value which provided highest accuracy.

We further compared the proposed scheme against previously published work in (Liu and Lee, 2015). To compare against (Liu and Lee, 2015), we applied the three algorithms EWMA, SVR and NN to the throughput time series to predict the future throughput. The accuracy of the results obtained are compared with our proposed model using MLP (using both data models) and the obtained MSE and MAE metrics as depicted in Table 7.

The analysis of the results show that the proposed scheme using MLP achieves far better accuracy than the method in (Liu and Lee, 2015). For instance, the proposed scheme using MLP (Model-I) achieves highest accuracy gain of 29% as compared to EWMA methods in (Liu and Lee, 2015). It is worthy to note that despite using ML algorithms (SVR and NN) over past throughput values to predict future throughput does not achieve sufficient accuracy (72% for SVR and 79% for NN). The proposed scheme using MLP over Model-II also achieves almost similar accuracy gains over other methods.

The reason for higher accuracy of the proposed scheme over the other methods, is the use of necessary features and the data models. The throughput prediction as stated earlier in Section 1 is a complex function that depends upon various parameters. Prediction based on only the past throughput does not provide sufficient knowledge to estimate future throughput due to the randomness of the function.

7. Conclusions and future work

In this paper, we propose to use a machine learning based model to accurately estimate the transmission throughput in Wi-Fi networks. The proposed approach has the advantage over analytical models which are based on several assumptions and pose poor performance when the assumptions are not met in real scenarios. On the other hand, the proposed scheme is based on machine learning algorithms, which can learn from the actual network data and thus exhibit the characteristics of the networks with high accuracy. To implement the proposed scheme, benchmark datasets are created from raw network traces collected from simulations of Wi-Fi network as well as real Wi-Fi network. We proposed two different data models, termed as Model-I and Model-II for feature extraction.

The performance of both data models is evaluated over the acquired datasets and results are compared, to quantify the significance of each data model. It was shown that Model-I outperforms Model-II in terms of error performance and exhibits better generalization. Furthermore, the complexity analysis shows that Model-I outperforms in the feature extraction stage by taking less time to extract features. Model-I is also computationally efficient in training and prediction stages. The shortcoming of Model-I is that the size of features set and computation time increases linearly by increasing number of stations in the network. The prediction performance for both data models is further investigated over four ML algorithms, where MLP showed the highest prediction accuracy and robustness, followed by DT, RF and SVR respectively. As future work, the work can be extended to predict other performance metrics in wireless networks such as packet loss, end-to-end delay and jitters. The integration of the proposed approach into software-defined-networks (SDN) based controllers for Wi-Fi networks using OpenFlow can be significant contribution of the this work.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the Qatar National Research Fund (a member of Qatar Foundation) under NPRP Grant 8-627-2-260.

References

Alliance, Wi-Fi, 2016. Wi-Fi Peer-To-Peer (P2P) Technical Specification, Version 1.7. Wi-Fi Alliance Technical Committee P2P Task Group.

Andrzej, M., Ratajczak, W., 1993. Principal components analysis (PCA). Comput. Geosci. 19 (3), 303–342.

Anwar, M.Z., Kaleem, Z., Jamalipour, A., 2019. Machine learning inspired sound-based amateur drone detection for public safety applications. IEEE Trans. Veh. Technol. 68 (3), 2526–2534.

Assra, A., Yang, J., Champagne, B., March 2016. An em approach for cooperative spectrum sensing in multiantenna cr networks. IEEE Trans. Veh. Technol. 65 (3), 1229–1243.

Bui, N., Michelinakis, F., Widmer, J., May 2014. A model for throughput prediction for mobile users. In: European Wireless 2014; 20th European Wireless Conference, pp. 1–6.

Chen, K., Huang, L., 2018. Timely-throughput optimal scheduling with prediction. IEEE/ACM Trans. Netw. 26 (6), 2457–2470.

Choi, K.W., Hossain, E., Feb 2013. Estimation of primary user parameters in cognitive radio systems via hidden markov model. IEEE Trans. Signal Process. 61 (3), 782–795.

Combs, Gerald, et al., 2008. Version 0.99. Wireshark-network Protocol Analyzer, vol. 5.

Donohoo, B.K., Ohlsen, C., Pasricha, S., Xiang, Y., Anderson, C., Aug 2014. Context-aware energy enhancements for smart mobile devices. IEEE Trans. Mob. Comput. 13 (8), 1720–1732.

Fadlullah, Z., Tang, F., Mao, B., Kato, N., Akashi, O., Inoue, T., Mizutani, K., 2017. State-of-the-art deep learning: evolving machine intelligence toward tomorrow's intelligent network traffic control systems. IEEE Commun. Surv. Tutor. PP (99), 1.

Feng, V., Chang, S., March 2012. Determination of wireless networks parameters through parallel hierarchical support vector machines. IEEE Trans. Parallel Distrib. Syst. 23 (3), 505–512.

Fontes, R.R., Afzal, S., Brito, S.H.B., Santos, M.A.S., Rothenberg, C.E., nov 2015. Mininet-WiFi: emulating software-defined wireless networks. In: Proceedings of the 11th International Conference on Network and Service Management, CNSM 2015. IEEE, pp. 384–389.

Guo, J., Yang, C., Chih-Lin, I., 2018. Exploiting future radio resources with end-to-end prediction by deep learning. IEEE Access 6, 75729–75747.

He, Qi, Dovrolis, Constantine, Ammar, Mostafa, 2005. On the predictability of large transfer TCP throughput. In: ACM SIGCOMM Computer Communication Review, vol. 35. ACM, pp. 145–156. no. 4.

Jacobson Van, Craig, Leres, McCanne, S., 1989. The Tcpdump Manual Page, vol. 143. Lawrence Berkeley Laboratory, Berkeley, CA.

Jiang, C., Zhang, H., Ren, Y., Han, Z., Chen, K.C., Hanzo, L., April 2017. Machine learning paradigms for next-generation wireless networks. IEEE Wirel. Commun. 24 (2), 98–105.

Kato, N., Fadlullah, Z.M., Mao, B., Tang, F., Akashi, O., Inoue, T., Mizutani, K., 2017. The deep learning vision for heterogeneous network traffic control: proposal, challenges, and future perspective. IEEE Wirel. Commun. 24 (3), 146–153.

Khan, M.A., Dec 2019. Datasets of Transmission throughput in Wireless LANs. Available at: https://data.world/engrasifkhan/wlan-throughput.

Khan, M.A., Cherif, W., Filali, F., Hamila, R., 2017. Wi-fi direct research-current status and future perspectives. J. Netw. Comput. Appl. 93, 245–258.

Khan, M.A., Hamila, R., Kiranyaz, M.S., Gabbouj, M., 2019. A novel uav-aided network architecture using wi-fi direct. IEEE Access 7, 67305–67318.

Kim, T.-H., Yang, Q., Lee, J.-H., Park, S.-G., Shin, Y.-S., 2007. A mobility management technique with simple handover prediction for 3g lte systems,. In: 2007 IEEE 66th Vehicular Technology Conference. IEEE, pp. 259–263.

Kousias, K., Alay, ., Argyriou, A., Lutu, A., Riegler, M., 2019. Estimating downlink throughput from end-user measurements in mobile broadband networks. In: 2019 IEEE 20th International Symposium on A World of Wireless, Mobile and Multimedia Networks(WoWMoM). IEEE, pp. 1–10.

Kurose, J.F., Ross, K.W., 2013. Computer Networking: a Top-Down Approach: International Edition. Pearson Higher Ed.

Liaw, A., Wiener, M., et al., 2002. Classification and regression by randomforest. R. News 2 (3), 18–22.

Liu, Y., Lee, J.Y.B., Dec 2015. An empirical study of throughput prediction in mobile data networks. In: 2015 IEEE Global Communications Conference (GLOBECOM), pp. 1–6.

M. Team, 2012. Mininet: an Instant Virtual Network on Your Laptop (Or Other PC).

Mirza, P.B.M., Sommers, J., Zhu, X., Aug 2010. A machine learning approach to TCP throughput prediction. IEEE/ACM Trans. Netw. 18 (4), 1026–1039.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.

Samba, A., Busnel, Y., Blanc, A., Dooze, P., Simon, G., 2016. Throughput prediction in cellular networks: experiments and preliminary results. In: 1res Rencontres Francophones sur la Conception de Protocoles, lvaluation de Performance et l'Exprimentation des Rseaux de Communication (CoRes 2016).

Sundaresan, Srikanth, Feamster, Nick, Teixeira, Renata, 2015. Measuring the performance of user traffic in home wireless networks. In: International Conference on Passive and Active Network Measurement. Springer, pp. 305–317.

Timofeev, R., 2004. Classification and Regression Trees (Cart) Theory and Applications. Humboldt University, Berlin.

Wei, B., Kanai, K., Katto, J., 2016. History-based throughput prediction with hidden markov model in mobile networks. In: 2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). IEEE, pp. 1–6.

Wen, C.K., Jin, S., Wong, K.K., Chen, J.C., Ting, P., March 2015. Channel estimation for massive mimo using Gaussian-mixture bayesian learning. IEEE Trans. Wirel. Commun. 14 (3), 1356–1368.

Xia, M., Owada, Y., Inoue, M., Harai, H.,Oct 2012. Optical and wireless hybrid access networks: design and optimization. IEEE/OSAJ. Opt. Commun. Netw. 4(10), 749–759.

Yue, C., Jin, R., Suh, K., Qin, Y., Wang, B., Wei, W., 2017. Linkforecast: cellular link bandwidth prediction in lte networks. IEEE Trans. Mob. Comput. 17 (7), 1582–1594.

Zahmatkesh, A., Kunz, T., 2017. Software defined multihop wireless networks: promises and challenges. J. Commun. Netw. 19 (6), 546–554.

Zhang, W., Zhang, Z., Chao, H., Guizani, M., June 2019. Toward intelligent network optimization in wireless networking: an auto-learning framework. IEEE Wirel. Commun. 26 (3), 76–82.

Muhammad Asif Khan received Bachelor of Science (BSc) and Master of Science (MSc) degrees in Telecommunication Engineering from University of Engineering and Technology (UET) Peshawar and University of Engineering and Technology (UET) Taxila, Pakistan in 2009 and 2013 respectively. He is currently studying Doctor of Philosophy (Ph.D.) in Electrical Engineering at Qatar University. In his Ph.D. research, he is working on novel methods to improve the QoS in Wireless LANs involving machine learning. He has been working as Researcher Assistant with Qatar Road Safety Studies Center (QRSSC - now called QTTSC) and Qatar Mobility Innovation Center (QMIC). He has been involved in several research projects funded by Qatar National Research Fund (QNRF), Ashghal Qatar and Maersk Oil. He has published several articles and conference papers. His current research interests include design and optimization of network protocols, machine learning, deep learning and Internet of Things (IoT). For more detailed information, please refer to: http://www.engrasifkhan.com.

Ridha Hamila received the Master of Science, Licentiate of Technology with distinction, and Doctor of Technology degrees from Tampere University of Technology (TUT), Tampere, Finland, in 1996, 1999, and 2002, respectively. Dr. Hamila is currently a Full Professor at the Department of Electrical Engineering, Qatar University, Qatar. From 1994 to 2002 he held various research and teaching positions at TUT within the Department of Information Technology, Finland. From 2002 to 2003 he was a System Specialist at Nokia research Center and Nokia Networks, Helsinki. From 2004 to 2009 he was with Emirates Telecommunications Corporation, UAE. Also, from 2004 to 2013 he was adjunct Professor at the Department of Communications Engineering, TUT. His current research interests include mobile and broadband wireless communication systems, Internet of Everything, and Machine Learning. In these areas, he has published over 150 journal and conference papers most of them in the peered reviewed IEEE publications, filed five US patents, and wrote numerous confidential industrial research reports. Dr. Hamila has been involved in several past and current industrial projects, Ooreedo, Qatar National Research Fund, Finnish Academy projects, EU research and education programs. He supervised a large number of under/graduate students and postdoctoral fellows. He organized many international workshops and conferences. He is a Senior Member of IEEE.

Nasser Ahmed Al-Emadi received the B.S. and M.S. degrees, both in electrical engineering, fromWestern Michigan University, Kalamazoo, in 1989 and 1994, respectively, and the Ph.D. degree from Michigan State University, East Lansing, in 1999. He is currently a Full Professor and Chairperson of the Department of Electrical Engineering, Qatar University, Qatar. His research interests include operation, planning, and control of power systems, artificial neural networks (ANN) and machine learning.

Serkan Kiranyaz was born in Turkey, 1972. He received his BS and MS degrees in Electrical and Electronics Department at Bilkent University, Ankara, Turkey, in 1994 and 1996, respectively. During 1996–2000, he worked as a Field Engineer in Schlumberger W&T and Senior Researcher in Nokia Research Center, Tampere, Finland. He received his PhD degree in 2005 and his Docency at 2007 from Tampere University of Technology, Institute of Signal Processing respectively. He was working as a Professor in Signal Processing Department in the same university during 2009–2015 and he held the Research Director position for the department and also for the Center for Visual Decision Informatics (CVDI) in Finland. He currently works as a Professor in Qatar University, Doha, Qatar. Prof. Kiranyaz has a noteworthy expertise and background in various signal processing domains. He published two books, more than 45 journal articles in several IEEE Transactions and other high impact journals and more than 100 papers in international conferences. He served as PI and LPI in several national and international projects. His principal research field is machine learning and signal processing. He is rigorously aiming for reinventing the ways in novel signal processing paradigms, enriching it with new approaches especially in machine intelligence, and revolutionizing the means of "IJlearn-to-process" I signals. This in turn allowed him to publish the two most popular articles in IEEE Transactions on Biomedical Engineering in the years 2010 and 2016 and to rank the 2nd place in the recent PhysioNet Grand Challenge among 48 international teams. In 2017, his research team has won the 1st place in this challenge among 75 international teams including the major companies and universities such as Philips Research, Philips Healthcare, Siemens, University of Oxford, EPFL, etc. His contributions in Computer Vision resulted the state-of-the art algorithm in automatic salient object detection, the Quantum Cuts, where he got the "IJBest Paper Award" advance the current state of the art in modelling and representation, targeting high longterm impact, while algorithmic, system level design and implementation issues target medium and long-term challenges for the next five to ten years. He, in particular, aims at investigating scientific questions and inventing cutting edge solutions in complex machine learning which is in one of the most dynamic areas where science combines with technology to produce efficient signal and information processing systems meeting the high expectation of the users. Dr. Kiranyaz is very familiar and active in the research communities in Finland, Turkey and Qatar as he spent long years abroad, both in Finland and Turkey where he built strong international collaboration network with reputed research teams. For more detailed information please refer to: http://qufaculty.qu.edu.qa/mkiranyaz/ and https://www.researchgate.net/ profile/Serkan_Kiranyaz

Moncef Gabbouj received his BS degree in electrical engineering in 1985 from Oklahoma State University, Stillwater, and his MS and PhD degrees in electrical engineering from Purdue University, West Lafayette, Indiana, in 1986 and 1989, respectively. Dr. Gabbouj is a Professor of Signal Processing at the Department of Signal Processing, Tampere University of Technology, Tampere, Finland. He was Academy of Finland Professor during 2011–2015. He held several visiting professorships at different universities. Dr. Gabbouj is currently the TUT-Site Director of the NSF IUCRC funded Center for Visual and Decision Informatics. His research interests include Big Data analytics, multimedia content-based analysis, indexing andretrieval, artificial intelligence, machine learning, pattern recognition, nonlinear signal and image processing and analysis, voice conversion, and video processing and coding. Dr. Gabbouj is a Fellow of the IEEE and member of the Academia Europaea and the Finnish Academy of Science and Letters. He is the past Chairman of the IEEE CAS TC on DSP and committee member of the IEEE Fourier Award for Signal Processing. He served as Distinguished Lecturer for the IEEE CASS. He served as associate editor and guest editor of many IEEE, and international journals. Dr. Gabbouj was the recipient of the 2017 Finnish Cultural Foundation for Art and Science Award, the 2015 TUT Foundation Grand Award, the 2012 Nokia Foundation Visiting Professor Award, the 2005 Nokia Foundation Recognition Award, and several Best Paper Awards. He published two books and over 700 journal and conference papers and supervised 45 doctoral and 58 Master theses.