# Naïve Bayes Supervised Learning based Physical Layer Authentication: Anti-Spoofing techniques for Industrial Radio Systems

**Andreas Weinand[1], Christoph Lipps[2], Michael Karrenbauer[1] and Hans Dieter Schotten[1,2]**
[1]TU Kaiserslautern, Institute for Wireless Communication and Navigation, Kaiserslautern, Germany
[2]German Research Center for Artificial Intelligence, Intelligent Network Research Group, Kaiserslautern, Germany

weinand@eit.uni-kl.de
Christoph.Lipps@dfki.de
karrenbauer@eit.uni-kl.de
schotten@eit.uni-kl.de, Hans_Dieter.Schotten@dfki.de

**Abstract:** Physical Layer Security (PLS) based authentication schemes are an alternative to conventional security schemes such as e.g. certificates or Message Authentication Codes (MACs). They can provide a more lightweight solution compared to traditional cryptography in order to meet the requirement of secure data transmission. However, errors in Physical Layer Authentication (PLA) techniques can occur due to adverse influences resulting from PLS schemes such as receiver noise or fading channels. Skillfull methods are therefore required in order to detect anormal system behaviour. One promising solution are supervised classification schemes. The application of Naïve Bayes (NB) classifiers for PLA is therefore proposed and evaluated within this work. Prior to that, we analyse the resource efficiency within typical Ultra Reliable Low Latency Communication (URLLC) applications and conduct a security overhead analysis. We propose strategies in order to overcome the problem of missing training data from either the URLLC user or attacker node. A real world Software Defined Radio (SDR) based testbed using OFDM (Orthogonal Frequency Division Multiplexing) is implemented in order to evaluate the performance of NB based PLA. The measurements are conducted within the german campus network frequency band (LTE band 43). Further, we conduct a hyperparameter optimization (HPO) based on random search. The investigated classifiers show promising results in terms of authentication accuracy and Receiver Operating Characteristic (ROC) curve performance.

**Keywords:** Physical Layer Authentication; Physical Layer Security; Machine Learning; Ultra Reliable Low Latency Communication; Machine Type Communication

## 1. Introduction

Many Ultra Reliable Low Latency Communication (URLLC) services, such as automated driving systems, telesurgery, factory automation and many more are currently getting enabled by technologies such as 5G-New Radio (NR). Therefore, new approaches regarding system reliability, availability and security need to be considered in order to meet the high requirements for this type of services. Conventional solutions have limitations with respect to system efficiency, since they were originally designed for large packet internet traffic such as Mobile Broadband (MBB). This is especially the case for solutions with respect to information security. In URLLC scenarios the traffic is however characterized by short and periodic messages which are transmitted at deterministic time slots (e.g. within the order of 10s of ms (Bockelmann, et al. 2017) (Yilmaz, et al. 2015)).

### 1.1 URLLC message security overhead ratio

In order to transmit cryptographically secure messages within an URLLC service, it is typically required to meet the security goals of message confidentiality, authentication and integrity. While confidentiality can be achieved by encryption of the message payload, authenticity and integrity are typically ensured by attaching authentication tags (e.g. Message Authentication Codes (MACs)) to messages. This leads however to an increase of the original payload length. For MACs the recommendation of the Internet Engineering Task Force (IETF) is to either use a cipher based MAC (CMAC), e.g. based on a Advanced Encryption Standard (AES)-128 block cipher, or a hash based MAC (HMAC) which is based on a cryptographic hash function. For AES-128 based CMAC a maximum shortening to 64 Bit is recommended (IETF, RFC 4493, 2006), while for HMAC a minimum MAC size of 80 Bit is recommended (IETF, RFC 2104, 1997). Additionally, the encryption process leads to a further increase of the message size due to zero-padding, which is typically not notable in broad band applications. For URLLC however this is not always the case if very short messages are used (e.g. within the order of 10s to 100s of Bits (Bockelmann, et al. 2017) (Yilmaz, et al. 2015)). Protection against replay attacks requires Further additional overhead (e.g. counters, time stamps). Figure 1 illustrates the overhead introduced by authentication tags, zero-padding and replay protection for different URLLC message sizes. The Security Overhead Ratio (SOR), which is

defined as the percentage of security overhead compared to the payload of a message depending on the URLLC payload of $N$ Bit, is then calculated as

$$\mathrm{SOR} = \frac{ceil\left(\frac{N}{B}\right)B+L+K}{N} - 1,$$

where $ceil()$ is the ceiling function (maximum of the two nearest integer numbers), $B$ the cipher block size (e.g. 128 Bit for AES-128 cipher) $L$ the length of the MAC tag in Bit and $K$ the length of the counter in Bit.

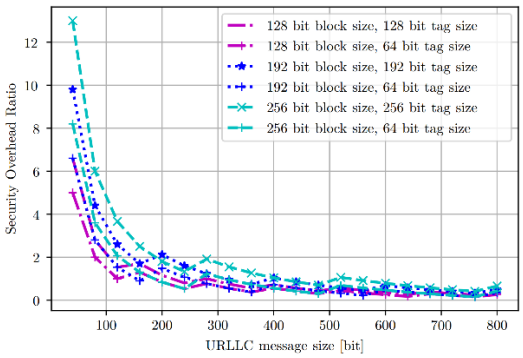



**Figure 1: SOR for confidentiality, integrity and replay protection**

The SOR is quite significant for short payloads and can reach up to 1000% of the URLLC message size. The SOR reaches 460% for a URLLC payload of 100 Bit, if a high level of security is applied (e.g. AES-256 cipher without MAC truncation and 48 Bit counter). The actual efficiency (application payload vs. overhead) on the air interface is even lower since this calculation is not considering the additional overheads introduced by non security functionalities. This work proposes therefore alternative solutions based on PLS for message authentication, integrity and replay protection in order to increase the ressource efficiency.

### 1.2 Organisation

The remainder of the work is organized as follows. In section 2 we give an overview of related work with respect to previously considered PLA approaches and in section 3 we introduce the system model. The NB classifiers which are used for Physical Layer Authentication (PLA) are presented within section 4. In section 5, we present techniques for artificial training data generation in case of the absence of an attacker during the training phase. Further, within sections 6 and 7 we present the evaluation methodology and results of the experimental evaluation respectively. Section 8 concludes the paper and provides an outlook for future work.

## 2. Related Work

Statistical methods were initially used for PLA, such as (Xiao, Greenstein, et al. 2007), where an approach based on simulation of the wireless channel and hypothesis testing is presented for static scenarios (no user mobility). It is later extended to mobile (time variant) scenarios in (Xiao, Greenstein, et al. 2008). The authors propose to use Neyman-Pearson Hypothesis Testing and least square adaptive filters in order to check if channel estimations of subsequent transmissions of a user are related to each other. The users they are considering have a low mobility (e.g. pedestrian speed) and they use an OFDM based transmission system for performance evaluation. The authors in (Tugnait und Kim 2010) propose a channel state information based authentication method for a single carrier system using whiteness of residuals testing. In (Baracca, Laurenti und Tomasin 2012), the authors consider a Multiple Input Multiple Output (MIMO) OFDM system and propose Generalized Likelihood Ratio Testing in order to detect attacks based on channel estimation data. They further introduce various attack strategies under which they evaluate the performance of their approach using Montecarlo simulations. They show how the misdetection probability is depending on different parameters such as the Signal-to-Noise Ratio (SNR), the number of subcarriers, the correlation coefficient, the probability of false alarms and knowledge of the attackers behaviour.

One of the first works that proposed machine learning (ML) based techniques in order to detect and identify transmitters based on physical layer metadata is (Gulati, et al. 2013), where the authors use Gaussian Mixture Models in order to detect an attacker. The feature they use are different antenna modes (angle of a leaky wave antenna) and they evaluate their proposed scheme within a static scenario at different parameter settings (number of different angles, attack probability, number of training messages, carrier frequency). Two approaches based on ML (Support Vector Machines and Linear Fisher Discriminant Analysis) are presented in

(Pei, et al. 2014). In (Forssell, et al. 2019), the authors use a single input single output system in order to detect the transmitter of data packets based on the respective line-of-sight components. The authors in (Weinand, et al. 2019) propose the application of supervised learning for channel based PLA within URLLC in a static environment. In (Wang, et al. 2017), the authors propose the usage of extreme learning machines for PLA. They further introduce a pseudo adversary model in order to generate artificial attacker training data. The authors in (Senigagliesi, Baldi und Gambi 2020) use Neural Networks and Support Vector Machines for PLA and evaluate the classifiers using a simulative framework. They addtionally propose to use one-class versions of the respective models in case of missing training data from the attacker node. Further, they conduct a model selection and manual HPO. They do however not consider cryptographic schemes in any phase of their proposed PLA procedure.

Though the results of most of the presented related works are promising, there are still several drawbacks existing. Many works use statistical decision metrics, which require a constant threshold to be set as input. This approach delivers typically only optimum performances in the cases which are evaluated. ML based classification schemes can therefore be applied in order to avoid that. Further, the outcome of simulative evaluations needs to be verified in real world scenarios with real hardware such as e.g. prototypes in order to prove their field performance. Cross validation should further be considered over a variety of classification schemes and parameterizations (e.g. HPO). Additionally, real world scenarios within the IR scope should be considered in order to pave the road towards standardization of PLA schemes within existing or future wireless systems. Channel measurements are therefore conducted within the 3.7 GHz campus network frequency band (Germany) within this work and realisitic URLLC traffic models are used in order to evaluate the PLA performance.

## 3. System Model

Within this section, we introduce the considered system model, as well as possible physical layer features to be used for PLA. Further, we introduce the considered attacker scenario.

### 3.1 Wireless System

We consider a wireless (public) channel between two users, Alice and Bob, intending to exchange authenticated messages with each other. A third user Eve, who is an adversarial user, tries to inject illegal messages masquerading as one of the legal users Alice or Bob. This can be done by reusing the radio ressources allocated to either Alice or Bob in case of centralized Radio Resource Management (RRM) or by spoofing their addresses in case of competetive/decentralized RRM. Figure 2 indicates the possible message flows over the wireless channel for all considered users. Messages transmitted by users are denoted with $\boldsymbol{x}_u$ and messages received by users with $\boldsymbol{y}_u$, where u denotes user Alice ($u = A$), Bob ($u = B$) or Eve ($u = E$).

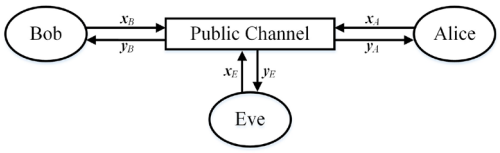



**Figure 2: System Model**

### 3.2 Physical Layer Metadata

Physical layer protocol metadata is used in PLS schemes in order to provide lightweight security functions. This metadata can either originate from the transceiver hardware (e.g. radio fingerprints or clock skew) or it can originate from characteristics of the wireless channel (e.g. RSSI or channel estimation data). In the first case it is already challenging for an attacker to spoof these characteristics by using default hardware (off-the-shelf). However, this requires an additional effort for the legitimate receiver in order to detect these characteristics and increases the receiver complexity and cost. For the second case, the respective data is typically available and accessible in any radio protocol. Therefore, we focus on channel based metadata and especially channel estimation data in our work, which is computed at the receiver side in form of a frequency domain channel estimation (OFDM based systems). The received signal at a user $u_i$ in such systems is

$$\boldsymbol{y}_{u_i} = \boldsymbol{H}_{u_i u_j} \boldsymbol{x}_{u_i} + \boldsymbol{n}_{u_i u_j} \quad (1)$$

with

$$\boldsymbol{H}_{u_i u_j} = \left[ h^0_{u_i u_j}, \dots, h^{M-1}_{u_i u_j} \right] \quad (2)$$

being the $M$-dimensional channel matrix of the frequency selective Single Input Single Output (SISO) fading channel between user $u_i$ and $u_j$ (transmitter) and $h^l_{u_i u_j}$ being the complex gain of the $l$-th sample in frequency domain ($l = 0, \dots, M-1$). It can be estimated as

$$\widehat{\boldsymbol{H}}_{u_i u_j} = \boldsymbol{H}_{u_i u_j} + \boldsymbol{\epsilon}_{u_i u_j} \quad (3)$$

by user $u_i$. Due to noise $n_{u_i u_j}$ which is modelled as a gaussian random variable with zero mean and variance $\boldsymbol{\sigma}^2_{n_{u_i u_j}}$ the channel estimation is not perfect and errors $\boldsymbol{\epsilon}_{u_i u_j}$ occur.

### 3.3 Attacker Model

A typical scenario for an attacker Eve is, that she is at a spatially different location compared to Bob and Alice (we assume a distance of more than the wavelength of the transmitted signal) and uses advanced equipment such as directed antennas and high sensitivity receivers in order to maximize the coverage to her benefits. We further assume perfect knowledge of the underlying communication protocol at Eve to run active attacks such as masquerade attacks, replay attacks or address spoofing attacks by introducing messages $\boldsymbol{x}_E$.

It is not assumed that Eve is gaining physical access to Alice or Bob to accomplish invasive attacks such as hardware modification. Further, other active attacks such as Denial-of-Service (DoS) attacks due to jamming are not considered. We assume that the legitimate communicating participants Bob and Alice have already carried out initial user authentication to each other and have set up trust in a secure way by using cryptographic certificates or MACs. Attacks on the initial authentication stage are not considered. Further, we assume that Eve is, depending on the current attack strategy, spoofing on transmission slots which are actually allocated to Bob by using a higher transmit power in order to make Alice overhear Bobs transmissions. In order to successfully carry out attacks, the receive power of Eves transmissions has to be higher than that of Bobs transmissions at Alice.

Within this work we consider that the attacker node is transmitting messages at a constant probability which is denoted Attack Intensity (AI). It gives the percentage of packets injected by Eve based on the total number of packets transmitted by Bob or Eve in total.

## 4. Naïve Bayes classification for PLA

After introducing the channel based PLA concept, we present the NB classifiers that were used in order to evaluated the PLA performance within our testbed.

### 4.1 Wireless channel based PLA

Within PLS based authentication such as PLA, receivers try to identify the transmitter based on e.g. channel characteristics instead of evaluating an authentication tag. Considering Bob transmitting messages $x_B(k)$ to Alice, Alice will estimate the channel as $\widehat{\boldsymbol{H}}_{AB}(k)$ when receiving message $\boldsymbol{y}_A(k)$ at times $k = 0, \dots, N_{train} - 1$. The messages $\boldsymbol{x}_B(k)$ sent by Bob need to contain cryptographic information from higher layers (above physical layer) in order to prohibit any spoofing attacks by Eve during time $k = 0, \dots, N_{train} - 1$. Alice authenticates Bob then using PLA from time $k \geq N_{train}$. This is realized by comparing the physical parameters such as channel state estimation or Channel Impulse Response (CIR) to the previously recorded versions at times $k = 0, \dots, N_{train} - 1$. Based on that, two hypothesis are possible about the origin of the respective estimated channel $\widehat{\boldsymbol{H}}(k), k \geq N_{train}$ and the payload of messages $y_A(k)$, which are

$$\mathcal{H}_0: \widehat{\boldsymbol{H}}(k) \text{ due to } \boldsymbol{x}_B(k) \text{ and}$$
$$\mathcal{H}_1: \widehat{\boldsymbol{H}}(k) \text{ due to } \boldsymbol{x}_E(k). \quad (4)$$

In case of $\mathcal{H}_0$ follows

$$\widehat{\boldsymbol{H}}(k) = \boldsymbol{H}_{AB}(k) + \boldsymbol{\epsilon}_{AB}(k) \quad (5)$$

and in case of $\mathcal{H}_1$

$$\widehat{\boldsymbol{H}}(k) = \boldsymbol{H}_{AE}(k) + \boldsymbol{\epsilon}_{AE}(k). \quad (6)$$

The reference channel estimation used for comparison needs to be updated due to temporal variations in wireless channels, such as due to doppler fading. Otherwise it is not possible to distinguish between channel variations and messages introduced by an attacker after some point. Therefore, the update interval should be below the channel coherence time $T_C$ of the $\boldsymbol{H}_{AB}$ channel in order to catch up with these variations. The temporal difference between two subsequent channel estimations should further always be below $T_C$. Due to

such infuences on the physical parameters used within the PLA, errors can occur. Therefore, we propose the utilization of NB supervised classification schemes in order to distinguish between transmissions of users Bob and Eve received by Alice respectively.

## 4.2 NB based supervised classification

The classification algorithms, which are based on NB methods, are applying Bayes' theorem. Given the value of the class variable it assumes the conditional independence between every pair of features. Based on $y$ as the target variable and the dependent feature vector $x = [x_1, \dots, x_n]$, Bayes' theorem states that

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1,\dots,x_n|y)}{P(x_1,\dots,x_n)}. \quad (7)$$

From the naive independence assumption $P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y)$ follows for all $i$

$$P(y|x_1, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1,\dots,x_n)}. \quad (8)$$

Due to the constant $P(x_1, \dots, x_n)$, the classification rule $P(y|x_1, \dots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$ can be derived, which further results in $y_{pred} = \arg\max_y P(y)\prod_{i=1}^{n} P(x_i|y)$. The terms $P(y)$ (frequency of class $y$ in the training set) and $P(x_i|y)$ can be estimated using Maximum A Posteriori (MAP) estimation. The term $P(x_i|y)$ however can be estimated in different ways based on the assumptions which are made on its distribution (Zhang 2004). Therefore, different NB approaches exist which are introduced below and implemented in our testbed.

### 4.2.1 Gaussian Naïve Bayes classifier

The Gaussian Naïve Bayes (GNB) classification algorithm estimates $P(x_i|y)$ as

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\rho_y^2}} e^{-\frac{(x_i-\mu_y)^2}{2\rho_y^2}} \quad (9)$$

assuming the likelihood of the features is gaussian. In order to estimate $\rho_y$ and $\mu_y$, maximum likelihood estimation is used.

### 4.2.2 Complement Naïve Bayes classifier

The Complement Naïve Bayes (CNB) algorithm is especially suited for imbalanced data sets. It uses statistics, which are derived from the complement of each class in order to compute the weights $w$. These are calculated as

$$w_{\tilde{c}i} = \log \hat{\theta}_{\tilde{c}i} = \frac{w_{ci}}{\sum_j |w_{ci}|} \text{ with } \hat{\theta}_{\tilde{c}i} = \frac{N_{\tilde{c}i} + \alpha_i}{N_{\tilde{c}} + \alpha} \quad (10)$$

Where $N_{\tilde{c}}$ is the total number of samples within the opposite class and $N_{\tilde{c}i}$ are the repetitions of a sample in the opposite class (McCallum und Nigam 1998). Further, $\alpha_i$ is a smoothing hyperparameter and $\alpha = \sum_i \alpha_i$. Based on that, the CNB classification rule is

$$\hat{c} = \arg\min_c \sum_i x_i w_{ci} \quad . (11)$$

### 4.2.3 Multinomial Naïve Bayes classifier

The Multinomial Naïve Bayes (MNB) classifier assumes multinomially distributed data, where the distribution is parametrized by vectors $\theta_y = [\theta_{y1}, \dots, \theta_{yn}]$ for each class $y$. $\theta_{yi}$ is the probability $P(x_i|y)$ of feature $i$ belonging to class $y$ and estimated according to

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n} \quad (12)$$

Here, $N_{yi} = \sum_{x \in T} x_i$ is the count of feature $i$ appearing in a sample of class $y$ within the training set $T$ and $N_y = \sum_{i=1}^{n} N_{yi}$ is the total count of all features for class $y$. A smoothed maximum likelihood estimation of $\hat{\theta}_{yi}$ is achieved by introduction of the smoothening parameter $\alpha \geq 0$, which allows for either Laplace smoothing ($\alpha = 0$) or Lidstone smoothing ($\alpha < 1$) (Rennie, et al. 2003).

## 5 Artificial Training Data Generation for PLA

A big issue of the practical application of PLA is the fact, that the attacker is typically not present during the training phase. Different strategies can be deployed in order to overcome this, where artificial (attacker) training data is generated. We propose the utilization of fixed sequences for that purpose, since they are in contrast to the approach in (Wang, et al. 2017) independent of the channel profiles of the URLLC users and can therefore be generated in a preliminary manner reducing complexity and cost. The possible options we investigate here are 0-sequences

$$h_{AE}^{i} = 0, i = 0, \dots, M-1, (13)$$

1-sequences

$$h_{AE}^{i} = 1, i = 0, \dots, M-1, (14)$$

and random sequences

$$h_{AE}^{i} = \text{rand}(0,1), i = 0, \dots, M-1, (15)$$

where $\text{rand}(0,1)$ is a random float number within the range of 0 and 1. We are further assuming normalized channel profiles of the URLLC users as described in section 6.1.

## 6 Evaluation Methodology

The evaluation methodology used within section 7 is presented in this section including steps such as data pre-processing, hyperparameter optimization and PLA performance metrics.

### 6.2 Data Pre-processing

The pre-processing steps as depicted within Figure 3 are applied to the validation and testing datasets respectively, in order to fit the classifiers. The magnitudes $\boldsymbol{F}(k) = \left|\widehat{\boldsymbol{H}}(k)\right|, k = 0, \dots, N$ of the complex channel gains are calculated first for each data sample $\widehat{\boldsymbol{H}}(k)$. A pre-processing step for feature dimensionality reduction is added in order to investigate the impact of feature dimensionality. This reduces the number of features of the original sample from $M = 48$ to $M_{red}$ by sampling the original feature vector at the respective rate yielding

$$\boldsymbol{F}_{red}(k, l^*) = \boldsymbol{F}\left(k, \frac{lM}{M_{red}}\right), \; l^* = 0, \dots, M_{red} - 1, (16)$$

where $l^*$ denotes the index of the features in the new feature dimension $M_{red}$ and $l$ the index in the original feature dimension.

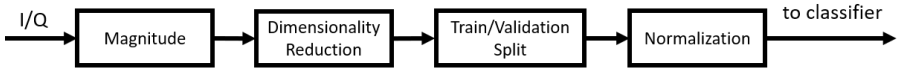



**Figure 3: Data pre-processing**

Afterwards, the data set is split into training and validation sets, which are used to fit the classifiers (training) and validate their performance. The training set is given by $\boldsymbol{F}_{train} = [\boldsymbol{F}_{red}(0), \dots, \boldsymbol{F}_{red}(N_{train} - 1)]$ and the remaining samples are used for validation. Finally, the training and validation data is normalized. This is done by calculating the mean and variance of the training set, then the training set is transformed to

$$\boldsymbol{F}_{train}^{norm} = \frac{\boldsymbol{F}_{train} - mean(\boldsymbol{F}_{train})}{std(\boldsymbol{F}_{train})} \; (17)$$

and the validation set is transformed according to

$$\boldsymbol{F}_{valid}^{norm} = \frac{\boldsymbol{F}_{valid} - mean(\boldsymbol{F}_{train})}{std(\boldsymbol{F}_{train})} \; . (18)$$

### 6.3 Hyperparameter Optimization

We use a random search in order to find the optimal hyperparameters

$$\lambda^* = \underset{\lambda \in \Lambda}{\operatorname{argmin}} \underset{x_k^{valid}}{\operatorname{mean}} L(y_k^{train}, f_\lambda(x_k^{train})) \quad (19)$$

to the given classification task. Possible value ranges for each model and respective hyperparameter are given as input and the models are fitted and validated on all combination of hyperparameters. For this, we only use $N_{valid}$ data sets of the overall $N_{data} = N_{valid} + N_{test}$ data sets. We use the accuracy metric in order to select the optimal hyperparameter set for each classifier. It is given by

$$P_A = \frac{1}{N} \sum_{k=N_{train}}^{N-1} \boldsymbol{y}_{pred}(k) - \boldsymbol{y}_{true}(k)\,, (20)$$

where $\boldsymbol{y}_{pred}$ is the label vector predicted by the respective classifier (user Bob or Eve is the transmitter) and $\boldsymbol{y}_{true}$ the true label vector. The hyperparameters derived from random search for the NB classifiers introduced in section 4.2 are listed in Table 1 accordingly. The remaining $N_{test}$ data sets are used in order to evaluate their performance on the given classification task after deriving the optimal hyperparameters for each classifier. Beside the accuracy metric, ROC curves are used including Area Under Curve (AUC) scores for that purpose.

**Table 1: Optimal hyperparameters for each classifier**

| Classifier | Parameter | Value |
|---|---|---|
| GNB | var_smoothing | $1.5189 \cdot 10^{-7}$ |
| CNB | alpha | 0.6213 |
| MNB | alpha | 0.5042 |

## 7 Performance Evaluation

In this section, we first describe the process of acquisition of channel estimation data. Then the evaluation scenario is presented as well as results and their discussion respectively.

### 7.2 Data Acquisition

We used Universal Software Radio Peripheral (USRP) B210 SDR platforms and an OFDM based transmission systems in order to obtain realistic channel estimation data. The OFDM system is implemented with GNURadio OFDM transmitter and receiver baseband processing blocks. These provide the functionality for channel estimation based on the Schmidl and Cox method (Schmidl und Cox 1997) using the respective preamble symbols. We use an OFDM setup with a Fast Fourier Transform (FFT) size of 64 with 48 active subcarriers. The cyclic prefix length is 16 samples, whereas the bandwidth is 10 MHz and the carrier frequency is set to 3.715 GHz. Each data packet consists of 2 preamble symbols, which are used for channel estimation and frequency offset estimation at the receiver and N data symbols. The data packets are transmitted with a periodicity of 10 ms, which yields us a vector of 48 complex symbols for each received OFDM packet (only the active subcarriers are considered for channel equalization).

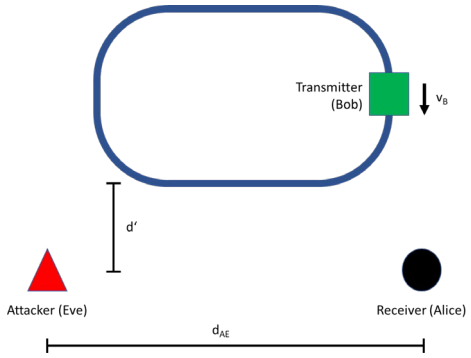



**Figure 4: Experiment setup with positions of Alice, Bob and Eve**

A mobile URLLC scenario is consider where user Bob is under mobility and emulating a URLLC user device transmitting messages, while Alice and Eve are static emulating the URLLC gateway receiving messages and the attacker node transmitting messages respectively (see **Figure 4**). The environment is a mixed office/lab area with a lot of objects and metal walls. Datasets of channel estimation data are recorded by Alice from $N_{data}$ different position constellations of the users (Alice and Eve). In case of a transmission of Eve, it is assumed, as mentioned in section 3.3, that Bobs regular transmission is interfered by Eve (e.g. using a comparatively higher transmit power). The experiment parameters are summarized within Table 2.

**Table 2: Validation and testing parameters**

| Parameter | Variable | Value |
|---|---|---|
| Carrier frequency | $fc$ | 3.715 GHz |
| Transmission frequency | $\Delta T$ | 100 Hz |
| Channel bandwidth | $BW$ | 10 MHz |
| No. of datasets | $N_{data}$ | 10 |
| No. of validation/test datasets | $N_{valid}/N_{test}$ | 5 |
| Feature dimension | $M$ | 48 |
| Reduction method | - | pilot |
| No. of training samples | $N_{train}$ | 10 |
| No. of samples per data set | $N$ | 10000 |
| User velocity | $v_B$ | $1\frac{m}{s}$ |

## 7.3 Results

The PLA performance of the NB classifiers is depicted in Figure 5 for different AIs. While the MNB and CNB have a more or less constant accuracy independent of the AI, the accuracy of the GNB decreases for low AIs. Such as at an AI of 20%, 99% accuracy is reached, whereas for AIs≥40%, the accuracy is ≥99.99% for all classifiers.

Within Figure 6, the ROC curves of NB classifiers are presented for low (a), medium (b) and high (c) AIs respectively. The PLA performance of the GNB is decreased here similarly for a low AI of 10% reaching an AUC score of 98.14%. While the AUC performance increases for the GNB with increasing AI to 99.56% (medium AI) and 99.47% (high AI) respectively, for the MNB and CNB a decrease of AUC performance is observed. Both are reaching AUC scores of 99.75% at medium AI and 99.06% at high AI.

The PLA performance in terms of accuracy depending on the feature dimension $M$ and training set size $N_{train}$ is depicted in Figure 7. In case of low AIs (Figure 7a), the PLA performance of the GNB is constant at 99.6% independent of M, while the performance of GNB and MNB is decreased for very low feature dimensions (M=4) reaching a minimum of 95.32%. However, for M=20, M=24 and M=48 it reached accuracies ≥99.98%. At medium AIs (Figure 7b), the minimum PLA performance reached 99.2% for the CNB and NMB classifiers and 99.9% for the GNB classifiers at M=4. For M≥16, all classifiers reach accuracies of ≥99.98%. A similar result is observed for high AIs (Figure 7c), where the minimum PLA performance reaches 99.85% at M=4 and 100% for M≥8 for all classifiers. In case of a decrease of the training set size, the PLA performance decreases for low (Figure 7d) and high (Figure 7f) AIs at low values of $N_{train}$. At $N_{train} = 50$ for instance, the performance of the GNB classifier reaches 94% and 98% respectively. In case of the GNB and MNB classifiers, a PLA performance of ≥99.8% is reached for Ntrain≥100 in this case. In case of a medium AI (Figure 7e) however, all classifiers reach a minimum performance of 99.99% in all cases.

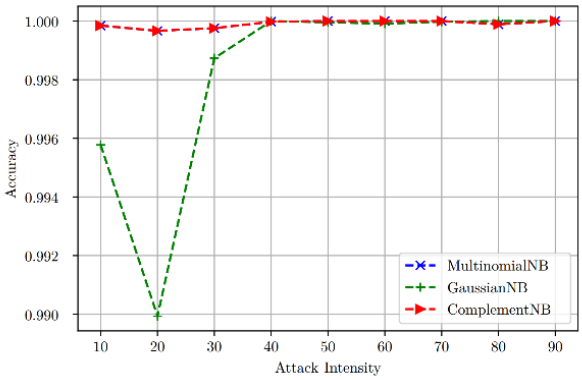



**Figure 5: Accuracy scores for NB classifiers vs. Attack Intensity**

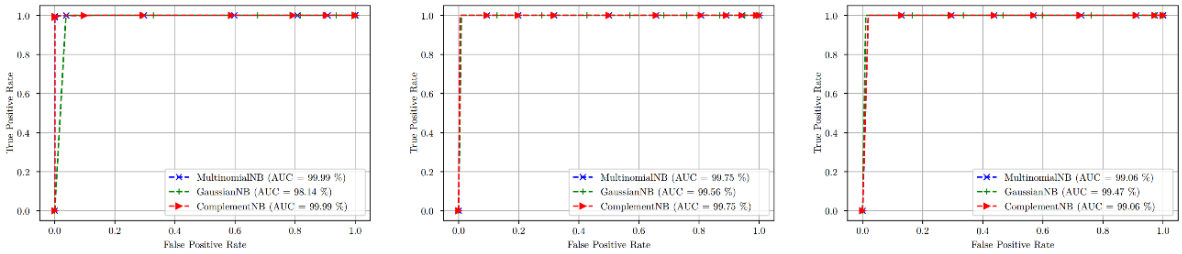
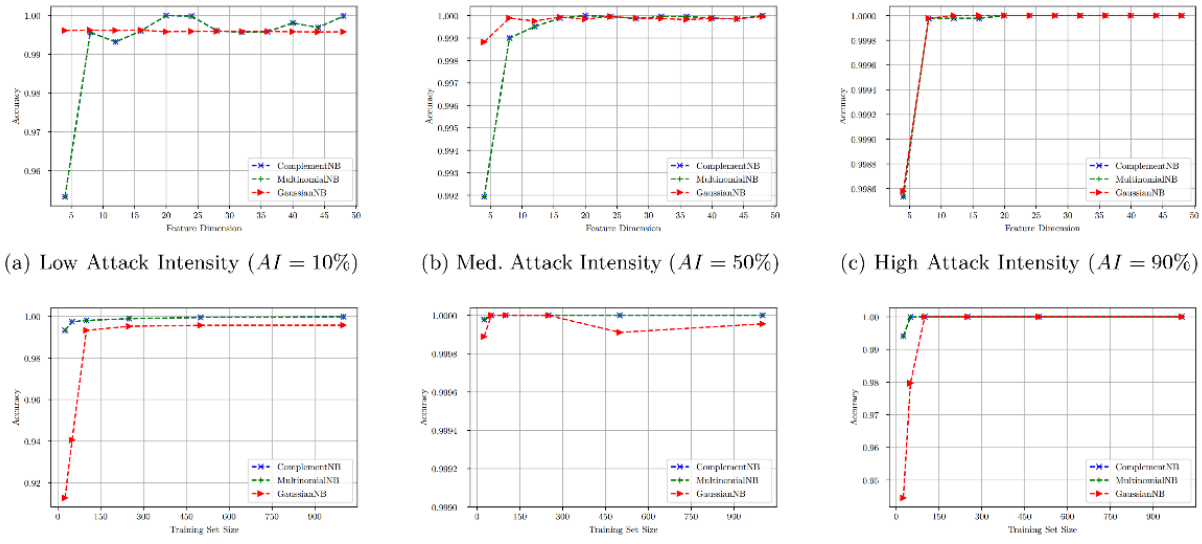
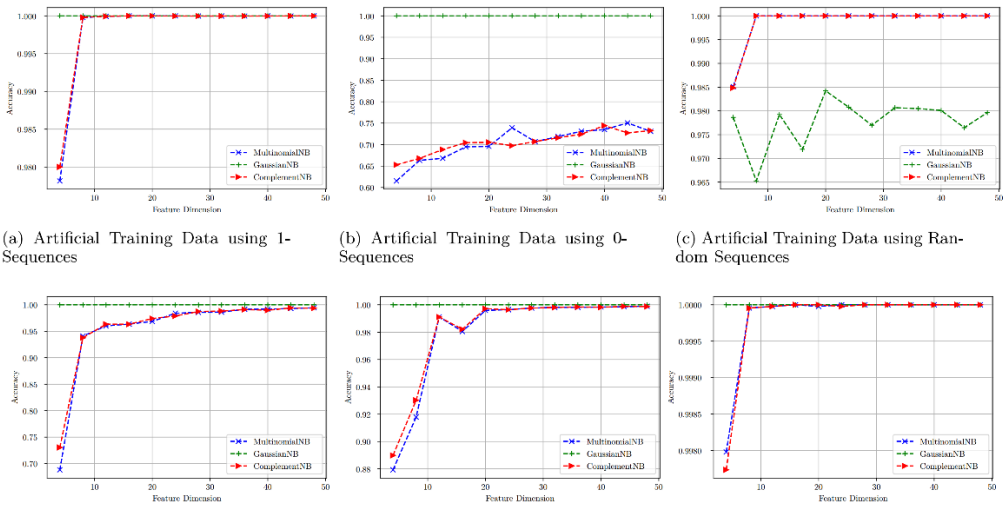



**Figure 6: ROC curves for NB classifiers under different Attack Intensities**




(a) Low Attack Intensity ($AI = 10\%$) (b) Med. Attack Intensity ($AI = 50\%$) (c) High Attack Intensity ($AI = 90\%$)

**Figure 7: Accuracy of NB classifiers vs. feature dimension (a)-(c) and training set size (d)-(f) under different Attack Intensities (low, medium, high)**




(a) Artificial Training Data using 1-Sequences

(b) Artificial Training Data using 0-Sequences

(c) Artificial Training Data using Random Sequences

**Figure 8: Accuracy of NB classifiers vs. feature dimension under different Artificial Training Strategies, AI = 0% (a-c) and AI = 100% (d-f)**

Figure 8 presents the results in case of missing training data from either Eve (AI=0%) or Bob (AI=100%). In case 1-sequences are used for artificial training data generation for Eves data (Figure 8a), the PLA performance is only decreased for low feature dimension (M=4) for the MNB and CNB classifiers at 97.2% and 98% respectively. In all other cases, the PLA performance is ≥99.98%. However, 0-sequences (Figure 8b) and random sequences (Figure 8c) provide suboptimal PLA performances for the MNB/CNB and GNB reaching a maximum of 73% and 98% at M=48 respectively. In case of missing training data from Bob, the PLA performance of the GNB is 100% independent of the used artificial training data generation method. However, in case of the MNB and CNB classifiers, the PLA performance decreases in case of a decrease in the feature dimension M. Minimum accuracies of 68.8-73% (1-sequences), 87.9-89% (0-sequences) and 99.77-99.8% (random sequences) are reached for instance at M=4. The maximum performance is reached at M=48 with accuracies of 99.36-99.45%, 99.87-99.9% and 100% respectively.

### 7.4 Discussion

The results presented in the previous subsection indicate a promising PLA performance in terms of the accuracy and AUC metrics. Further, the performance of the NB classifiers only decreases in rare scenarios (e.g. very low feature dimension or training set size). Especially the application of artificial training data strategies shows promising results in most of the cases. The choice of the respective method is however depending on the used classification scheme. In summary, the PLA performance derived within this work is similar or slightly above the performance reached in related work as presented in section 2.

## 8 Conclusion and Future Work

We have introduced the concept of PLA for industrial applications such as URLLC within this work. Further, we have proposed the usage of NB supervised classification for that purpose and presented methods in order to overcome the drawback of missing training data from respective users or attacker nodes in practical scenarios. The results derived from our experimental testbed show that under specific circumstances the application of NB supervised classification algorithms for PLA is a promising alternative to conventional security measures such as MACs. A considerable amount of radio resources can be saved within URLLC applications in that case. More advanced attack strategies such as e.g. burst attacks instead of attackers transmitting at a constant probability have to be considered within future work in order to further approve the robustness of PLA schemes.

## Acknowledgement

A part of this work was funded by the Federal Ministry of Education and Research (BMBF) of the Federal Republic of Germany in the research project “Industrial Radio Lab Germany” (grant number 16KIS1011).

## References


Baracca, P., N. Laurenti, und S. Tomasin. „Physical Layer Authentication over MIMO Fading Wiretap Channels." *IEEE Transactions on Wireless Communications*, July 2012.

Bockelmann, C., et al. „HiFlecs: Innovative Technologies for Low-Latency Wireless Closed- Loop Industrial Automation Systems." *22. VDE-ITG-Fachtagung Mobilkommunikation.* 2017.

Forssell, H., R. Thobaben, H. Al-Zubaidy, und J. Gross. „Physical Layer Authentication in Mission-Critical MTC Networks: A Security and Delay Performance Analysis." *IEEE Journal on Selected Areas in Communications*, 2019.

Gulati, N., R. Greenstadt, K. R. Dandekar, und J. M. Walsh. „GMM Based Semi-Supervised Learning for Channel-Based Authentication Scheme." *IEEE Vehicular Technology Conference (VTC Fall).* 2013.

IETF. „RFC 2104, HMAC: Keyed-Hashing for Message Authentication." 1997.

IETF. „RFC 4493, The AES-CMAC Algorithm." 2006.

McCallum, Andrew, und Kamal Nigam. „A comparison of event models for Naive Bayes text classification." *A comparison of event models for Naive Bayes text classification.* 1998.

Pei, C., N. Zhang, X. S. Shen, und J. W. Mark. „Channel-based physical layer authentication." *IEEE Global Communications Conference (GLOBECOM).* 2014.

Rennie, J. D. M., L. Shih, J. Teevan, und D. R. Karger. „Tackling the Poor Assumptions of Naive Bayes Text Classifiers." *Proceedings of the Twentieth International Conference on International Conference on Machine Learning.* Washington, DC, 2003.

Schmidl, T. M., und D. C. Cox. „Robust frequency and timing synchronization for OFDM." *IEEE transactions on communications*, 1997.

Senigagliesi, L., M. Baldi, und E. Gambi. „Comparison of Statistical and Machine Learning Techniques for Physical Layer Authentication." *IEEE Transactions on Information Forensics and Security*, 2020.

Tugnait, J. K., und H. Kim. „A channel-based hypothesis testing approach to enhance user authentication in wireless networks." *International Conference on COMmunication Systems and NETworks (COMSNETS 2010).* 2010.

Wang, N., T. Jiang, S. Lv, und L. Xiao. „Physical-Layer Authentication Based on Extreme Learning Machine." *IEEE Communications Letters*, 2017.

Weinand, A., R. Sattiraju, M. Karrenbauer, und H. D. Schotten. „Supervised Learning for Physical Layer Based Message Authentication in URLLC Scenarios." *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall).* 2019.

Xiao, L., L. Greenstein, N. Mandayam, und W. Trappe. „A Physical-Layer Technique to Enhance Authentication for Mobile Terminals." *2008 IEEE International Conference on Communications.* 2008.

Xiao, L., L. Greenstein, N. Mandayam, und W. Trappe. „Fingerprints in the Ether: Using the Physical Layer for Wireless Authentication." *2007 IEEE International Conference on Communications.* 2007.

Yilmaz, O. N. C., Y.-P. E. Wang, N. A. Johansson, N. Brahmi, S. A. Ashraf, und J. Sachs. „Analysis of ultra-reliable and low-latency 5G communication for a factory automation use case." *2015 IEEE International Conference on Communication Workshop (ICCW).* London, 2015.

Zhang, H. „The Optimality of Naïve Bayes." *In FLAIRS2004 conference.* 2004.