# Towards a Progressive E-health Application Framework

Thesis by

Zhirui Lu

In Partial Fulfillment of the Requirements

For the Degree of

Masters of Science

King Abdullah University of Science and Technology

Thuwal, Kingdom of Saudi Arabia

April, 2022

## EXAMINATION COMMITTEE PAGE

The thesis of Zhirui Lu is approved by the examination committee

Committee Chairperson: Prof. Basem Shihada
Committee Members: Prof. Ahmed Eltawil, Prof. Marco Canini

## ABSTRACT

Towards a Progressive E-health Application Framework
Zhirui Lu

Recent technological advances have opened many new possibilities for health applications. Next generation of networks allows real-time monitoring, collaboration, and diagnosis. Machine Learning and Deep Learning enable modeling and understanding complex and enormous datasets. Yet all the innovations also pose new challenges to application designers and maintainers. To deliver high standard e-health services while following regulations, Quality of Service requirements need to be fulfilled, high accuracy needs to be archived, let along all the security defenses to protect sensitive data from leaking.

In this thesis, we present a collection of works towards a progressive framework for building secure, responsive, and intelligent e-health applications, focusing on three major components, Analyze, Acquire, and Authenticate. The framework is progressive, as it can be applied to various architectures, growing with the project and adapting to its needs. For newer decentralized applications that perform data analysis locally on users' devices, powerful models outperforming existing solutions can be built using Deep Learning, while Federated Learning provides further privacy guarantee against data leakage, as shown in the case of sleep stage prediction task using smart watch data. For traditional centralized applications performing complex computations on the cloud or on-premise clusters, to provide Quality of Service guarantees for the data acquisition process in a sensor network, a delay estimation model based on queueing theory is proposed and verified using simulation. We also explore the novel idea of using molecular communication for authentication, named

Molecular Key, enabling the incorporation of environmental information into security policy. We envision this framework can provide stepping stones for future e-health applications.

# ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor Prof. Basem Shihada for his continuous support throughout my master journey. The completion of the thesis would not have been possible without his patience, encouragement, and immense knowledge. His optimism and vision have deeply inspired me, not only in research but also in life. It was a great pleasure and honor to have worked and studied under his guidance, and I am extremely grateful for this opportunity.

Beside my advisor, I would like to extend my sincere thanks to the rest of my thesis committee: Prof. Ahmed Eltawil and Prof. Marco Canini, for their insightful comments and thought-provoking questions. Special thanks to Dr. Osama Amin for his inspirational collaboration and enlightening discussions on the Molecular Key project.

I am grateful to my fellow colleagues and friends in the Networking Lab: Liang Zhang, Guoqing Ma, Wiem Abderrahim, and Sahar Ammar, for their helpful advice and practical suggestions during my research. Also, I would like to thank my friends: Feng Liang, Yuchen Li, Xi Peng, Ziwei Yang and Jihao Xin for their emotional support during the hardest moments.

Last but not least, I would like to thank my family for their love, care, and understanding. It is their faith in me that keeps driving me forward.

# TABLE OF CONTENTS

# LIST OF ABBREVIATIONS

| | |
|---|---|
| 3GPP | The 3rd Generation Partnership Project |
| Adam | Adaptive Moment Estimation |
| AUC | Area Under Curve |
| CNN | Convolutional Neural Network |
| CT | Computed Tomography |
| DL | Deep Learning |
| ECG | Electrocardiogram |
| EHR | Electronic Health Record |
| EM | Electromagnetic |
| FL | Federated Learning |
| FN | False Negative |
| FP | False Positive |
| GPS | Global Positioning System |
| IoT | Internet of Things |
| LSTM | Long-Short Term Memory |
| MC | Molecular Communication |
| ML | Machine Learning |
| MLP | Multi-Layer Perceptron |
| OTP | One-Time Password |
| PID | Photoionization Detector |
| PIN | Personal Identification Number |
| PSG | polysomnogram |
| QCI | QoS Class Identifier |
| QoS | Quality of Service |
| REM | Rapid Eye Movement |
| RFID | Radio Frequency IDentification |
| RMSE | Root-mean-square Error |
| RNN | Recurrent Neural Network |
| RPM | Remote Patient Monitoring |
| SGD | Stochastic Gradient Descent |

| | |
|---|---|
| SLA | Service-Level Agreement |
| SMS | Short Message Service |
| TP | True Positive |
| VOC | Volatile Organic Compound |
| WHO | World Health Organization |
| WSN | Wireless Sensor Network |

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1

# Introduction

The progress of the health industry has always been tightly connected to the advance of technology. With more versatile monitoring systems, more responsive network infrastructures, more powerful computing devices and more intelligent algorithms and models, e-health applications capable of providing personalized and precise medical services have been made possible.

On the monitoring stage, consumer grade wearable devices like smart bands and smartwatches enable vital metrics and motor activities to be recorded and analyzed continuously. Such fine granularity data is not only useful for long-term retrospective evaluation for doctors, but also allows critical conditions including heart failures and hard falling to be recognized and acted on appropriately as early as possible. Thanks to their availability, convenience and versatility, devices like Apple Watch empower the science community to gain further insight on areas that traditionally require complex and costly monitoring setups, including seizure[1], tremor[2], Parkinson's Disease[3] and sleep disorders[4]. Beside making direct contact, devices that utilize radar, lidar and depth sensors can even take measurements from a distance. These non-invasive monitoring methods reduce the learning cost for users and improves the data quality by minimizing the user interaction.

On the transmission stage, health applications are benefited from the ongoing infrastructure upgrade created by the demand of Internet of Things (IoT) devices. With the wide application of 4G and fiber network, as well as gradual rollout of 5G, low latency and high throughput transmission of health information is turning into reality, enabling real-time distant monitoring, collaborative diagnosis and remote surgery. This shortens the resource gap between different areas, making medical resources more accessible to everyone. For example, elders living in the country can still be monitored and receive medical suggestions from their healthcare provider as if they live in the city. In addition to long distance transmissions, local short distance networking is also seeing development, which allows smartphone and personal computers to act as a health information hub, aggregating readings from ambient, wearable and implantable sensors and relay them to remote servers located in hospitals for further analysis.

On the analysis stage, versatile data mining techniques including machine learning and deep learning can find hidden patterns and create predictive models automatically, even from the most complex and enormous datasets. They have seen remarkable success in various medical fields, from interpreting X-ray and Computed Tomography (CT) scan images for cancers [5] to modelling how proteins fold for drug development [6], while new models being developed for more complex problems and existing models being refined constantly for better accuracy. Beyond large and computation hungry models living in the server of hospitals, small and efficient models can also be created to run on the edge devices for lower latency and better personalization.

However, all these innovations also pose new challenges to both e-health application designers as well as maintainers aiming to provide high standard e-health services. On the transmission stage, even with modern network infrastructures, various factors such as routing and network usage may still affect the latency of health information being transmitted from users' sensor to the backend server, resulting in the violation of Quality of Service (QoS) requirements. On the analysis stage, building a model that aces at a specific problem is not simple, involving various design decisions and training details. Also, datasets the model trained on also limit its ability, yet sharing patients' data between hospital boundaries to create larger datasets may cause regu-

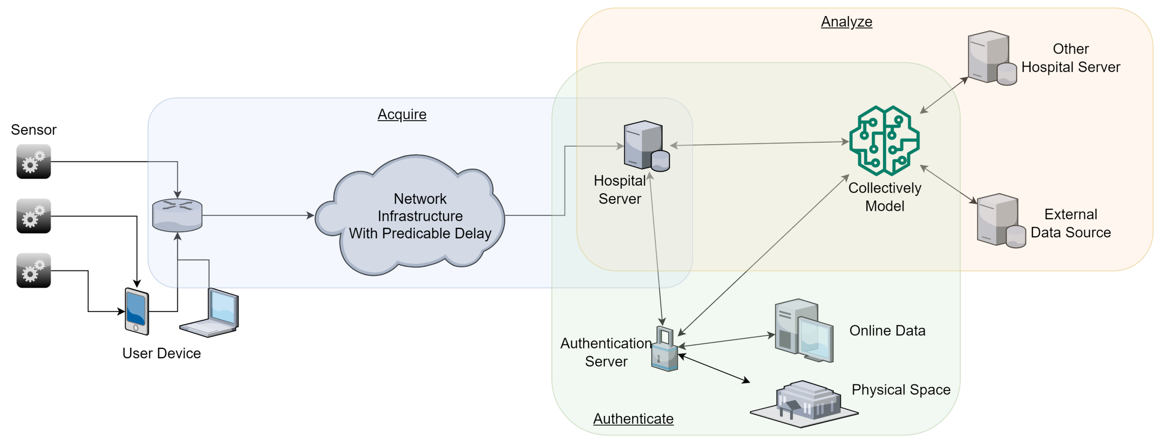

Figure 1.1: Framework overview

latory issues. On the security perspective, due to the private and sensitive nature of medical data, protecting the data and the analysis results derived against unauthorized access and adversarial attacks is also of great importance. Moreover, as more and more systems are evolving into hybrid systems that have both online and offline sections, it is now more important than ever to keep everyone safe, both virtually and physically under the current COVID pandemic.

In this thesis, we presented a collection of works towards a progressive framework for building secure, responsive and intelligent e-health applications, attempting to solve the challenges discussed above. We focused on 3 major components: Analyze, Acquire, and Authenticate, covering the whole lifecycle of data in e-health applications, as shown in Fig. 1.1. Unlike other frameworks that require a full-scale opt-in, our framework is progressive, as it provides options for applications at different stages and using different paradigms, whether the newer decentralized or traditional centralized, and is able to grow with the project, adapting to its needs. The framework establishes the following guidelines for building e-health applications:

- Service quality guarantee using mathematical models
- Unleash data collection potential from users' existing underutilized devices
- Data driven medical decision support using Machine Learning (ML) and Deep

Learning (DL)

- Privacy-preserving model training with Federated Learning (FL)

For newly built decentralized applications, we recommend the usage of deep learning for more accurate models for data analysis and federated learning to provide extra privacy guarantee, as shown in the Analyze chapter with the case of sleep stage prediction task using smart watch data. For traditional centralized applications, our framework provides the ability to estimate delay in a complex sensor network using models based on queueing theory and supported with simulation in the Acquire chapter, finding out the maximum intermediate nodes in a sensor network that satisfies the QoS requirements. In addition to existing security components, we also explored the possibility of using Molecular Keys based on Molecular Communication (MC) for authentication in the Authenticate chapter, creating an alternative authentication method that considers environmental information to be used along existing ones and expanding security beyond digital space.

# Chapter 2

# Background & Related Works

This chapter summarizes the background and related works of the topics discussed in the thesis.

## 2.1 E-health

This section introduces the concept of e-health, provides a categorization of e-health systems, and discusses the current focuses and challenges of e-health research.

### 2.1.1 E-health and variants

Since the introduction of the Internet, researchers have been trying to take advantage of the whole new possibility of transmitting information across political boundaries to improve the quality of healthcare services. In these early explorations, the word "e-health" is created, yet it lacks a formal definition and suffers from ambiguity, and all researches related to Internet and Healthcare are using it. The most widely accepted definition of e-health [7] defines it as "an emerging field in the intersection of medical informatics, public health and business", in which "health services and information delivered or enhanced through the Internet and related technologies". Moreover, in addition to related fields and technologies, the authors also elaborated on the meaning of "e" in e-health to describe its vision and expectations, including efficiency, empowerment, enabling, ethics and many others. In their work, they stated that e-health is much more than just technological advancements, but also represents a new way of thinking and framing healthcare utilizing new information and communication technologies.

In 1991, the second generation (2G) cellular network first introduced the ability of transmitting data directly to mobile phones. Later the third generation (3G) and fourth generation (4G) mobile telecommunications made further improvements on both the bandwidth and latency of data transmission. All the developments enabled a new era of integrating mobile phones into the e-health framework, as a client for receiving notification and alerts, or as a gateway of relaying sensor measurements to other components. Inspired by the new possibility, “m-health” is created as a subfield of e-health, defined as “The use of mobile and wireless technologies to support the achievement of health objects”. [8] M-health solutions will use not only the core cell network communication capacities of mobile devices, but also other onboard functionalities such as Global Positioning System (GPS) and Bluetooth. In World Health Organization (WHO)’s 2011 survey, they found that the most common m-health initiatives include health call centers, emergency toll-free services, emergency and disaster management and mobile telemedicine, and showed significant progress in improving the accessibility of healthcare services in traditionally underserved regions. Although innovations on utilizing data capability of mobile devices are being explored, many of the initiatives still use voice as the main communication method. With the gradual rollout of the fifth generation (5G) telecommunication services and the further expansion of IoT, it can be expected that the great potential of m-health on enhancing the responsiveness and coverage of healthcare for all will be unleashed.

In the healthcare industry, customizing personalized treatment plans tailored for a specific patient from general guidelines and therapeutic protocols has been a growing trend. However, the limited observability to health indicators outside of hospitals restricts the scope of personalization. Doctors struggle to understand the effect of a specific treatment on the daily life of patients, with the patient’s self-reported evaluation as the only data source. This issue is more severe in treating mental health issues and chronic diseases, both of which may be heavily influenced by the environment the patient is in, and the patient may not be able to record every fluctuation of their diseases. Thanks to the wide adoption of consumer-grade wearable sensors and global coverage of communication infrastructures, it is now possible to provide medical works with data collected during the interval between periodical visits. Recent development of machine learning and deep learning and many successful applications in the field of medicine also showed great potential of building useful models from large datasets and refine the models for specific cases. In [9], the authors proposed “i-health” as the next stage of e-health, in which “i” stands for both intelligent and personalized. They identified 3 key guidelines of i-health systems, including using new technologies for self-monitoring, extending assessment to cover patients’ environment and relationship, and utilizing data processing and analytic technologies for making personalized medical decisions. In their outlook, i-health will improve the effectiveness and precision of medical related decision making, creating a more personalized and tailored medical experience.

In summary, e-health is a broad term describing everything in the intersection of healthcare and the Internet. m-health is a subfield and a component of e-health, emphasizing on the usage of phones as a prominent element in e-health solutions. i-health is anticipated as the next stage of e-health, which will utilize new monitoring and data processing technologies to make healthcare more personalized and intelligent.

### 2.1.2 System Categorization

Based on the target user, maintainer and purpose, e-health systems can be classified into the following categories, as shown in Table 2.1.

Table 2.1: E-health System Categorization

| Target User | Maintainer / Operator | Purpose | Example Systems |
|---|---|---|---|
| Hospital / Healthcare Institute | Hospital / Healthcare Institute | Facilitate information exchange internally | - EHR System<br>- Specialty Information Systems (Imaging, Radiology, Pharmacy)<br>- Information Archival System<br>- Decision Support System<br>- Health Information Exchange |
| Patient | Hospital / Healthcare Institute | Provide healthcare service beyond physical boundaries | - Telehealth (Video/Audio/Text Consultation)<br>- Remote Patient Monitoring<br>- E-Prescribing<br>- Chronic Care Management |
| General User | App Developer / Service Provider / Device Manufacturer | Personal health monitoring and improvement | - Fitness App<br>- Personal Health Monitoring Service (run by device manufacturer) |

### 2.1.3 Research Focuses

Here is a simple summary of the current research focuses on the field of e-health.

**Electronic Health Record (EHR)** EHR is the collection of digitalized patients' health records, including but not limited to diagnoses, prescriptions, radiology images and laboratory results. The digital format allows the records to be read and written by machines, enabling it to be constantly updated to reflect the newest situation, transmitted and exchanged across multiple organizations when the patient moves. It also eases the building of case datasets and helps automating workflows, improving the efficiency and productivity overall. Though there are already existing standards, efforts are still being made to include data from more versatile sources.

**Remote Patient Monitoring (RPM)** Traditionally only in hospital patients can be constantly monitored, limiting their mobility. RPM uses a combination of sensor, communication and data process technologies to monitor patients' health outside of the clinical setting, creating a continuous stream of patients vital data to be transmitted to servers located in medical facilities. Beside data recording, some RPM systems have the capability to identify health risks and abnormalities automatically, then send alarms to patients and their caregiver for timely response.

**Wearable sensors** With the miniaturization advancements of sensor technologies, wearable consumer-grade electronic monitoring devices are now more capable, connected and user-friendly than ever before. Fitness trackers (e.g. Fitbit) and smart watches (e.g. Apple Watch) can provide measurements of vital signs including heart rate as well as exercise data such as step count and acceleration, while more specified devices such as wearable ECG monitors, blood pressure sensors and blood sugar sensors to cover specialized monitoring requirements. Data collected can be easily transmitted to users' smartphones wirelessly, and can be used for standalone personal health monitoring purposes, or transmitted to servers of medical institutes for further analysis in other e-health systems.

**5G & IoT** Though 5G is still being rolled out gradually, its promise of sub-millisecond latency and gigabyte bandwidth can surely be advantageous and transformative for various e-health applications, including remote monitoring, cross-institutional collaboration and virtual consultations. Internet of Things (IoT) allows more smart devices to be connected to the Internet and interact with each other, and e-health applications can benefit from inclusion of ambient sensors, enhanced connectivity by creating mesh networks and faster response utilizing computational powers at the edge and close to the user.

**ML / DL / FL** Machine learning made it possible to find patterns and build useful models given large datasets, and has already been extensively applied to various medical tasks. Deep learning can automatically build better features than manual tuning and utilize complex layers to create powerful models, and has seen successful usages in medical image segmentation and data extraction from unstructured medical records. However, as the inner workings of Deep Learning is still being discovered, currently the lack of interpretability and the risk of catastrophic forgetting restricts its potential in more life-critical tasks. Federated learning was first established as a privacy-preserving distributed learning scheme, making it a perfect fit for e-health application due to security and privacy regulations of medical data, yet it still suffers from possible efficiency and performance degradations compared to models trained in a centralized manner.

**COVID related** The ongoing COVID-19 pandemic has created many public health service challenges. Examples include contact tracing, rapid case identification and prompt notification, remote monitoring and support for people in quarantine, raising public awareness, improving transparency and many more. Researchers have been developing novel solutions to address them, such as location data and near field communication based contact tracing services, symptom reporting apps, diagnostic devices with IoT connectivity, virtual consultation platforms and visualization-supported data dashboards. [10]

### 2.1.4 Challenges of e-health Systems

**Interoperability** With more players including healthcare providers, application creators and sensor manufacturers entering the e-health field, ensuring data is interoperable between all participants is becoming increasingly challenging. Though standard formats are established for EHRs, there is still a lack of simple and secure patient data sharing methods. Even using the same format, the level of support still diverges between different applications. In contrast to the complex status quo, the majority of physicians expect for a more interoperable future, believing that improved interoperability can cut diagnosis time and improve patient outcomes according to Google's survey [11].

**Network & Data Transmission** Though Wi-Fi networks are getting more common and the coverage of cellular networks is almost ubiquitous and still constantly improving, transmitting data streams across networks is still not trivial. Users may accidentally walk into a signal blind spot and lose connection. When the network is congested the quality of service is hardly guaranteed. Handover between access points and interference from other devices and appliances may result in intermittent disconnection and bad user experience for time-sensitive applications. Possible solutions are being developed, including intelligent fallback to other available networks when the primary network is having problems, as well as build mesh networks and use devices that are still connected to relay data.

**Sensor** Sensors are literally the eyes of e-health systems, but there is still a long way for them to be perfect. One issue is accuracy, with some sensors requiring periodic calibration to function properly. Another issue is battery life. When a sensor is working, it has to consume power for making measurements, performing processing and transmitting data, resulting in the need for frequent recharging. Moreover, there are issues for pricing and maintenance.

**Complexity & Scalability** For most e-health applications, flux of data is continuously generated and transmitted, which then needs to be efficiently stored and analyzed. Although experience gained from building big data time-series applications can provide some guidance, these systems are still of great complexity, and may struggle to scale when the user base expands, new institutes getting involved or the uptime keeps growing. In addition, the critical nature of e-health applications enforces a strict Service-Level Agreement (SLA) and the downtime needs to be minimized, making the system design more challenging. One possible solution is to utilize edge computing and fog computing to offload some load to the network edge, yet this may increase the overall hardware cost and reduce accessibility.

**Security, Privacy & Regulatory Compliance** Due to the sensitive nature of medical data, the security of e-health systems is always one of the top priorities, and the design and operation needs to comply with regulations. In transmit data should be encrypted to prevent eavesdropping. Authentication and authorization policies are necessities to stop adversival attacks, along with an audit component to log all the intended accesses. Beside stopping outsiders from retrieving data, encapsulation and isolation are also necessary to avoid data belonging to one user being read by another. When interacting with external systems, data may need to be anonymized or added noise to protect privacy.

## 2.2 Remote Patient Monitoring

Remote Patient Monitoring can monitor the patient even when the patient is out of hospital, which has been shown to reduce readmission, length of hospital stay and emergency presentation. [12] Current RPM system can be applied to monitoring of various domains, including but not limited to heart disease, diabetes, mobility issues and mental health. [13]

### 2.2.1 System Components

A typical RPM system has the following components:

- Sensors: A series of sensors, fixed, wearable or ambient, used to monitor the patient's status as well as the environment.
- Gateway: Most sensors are not directly connected to the Internet, therefore gateway devices are needed to relay the data collected from sensors. In addition to relaying data, the gateway may also perform filtering, compression or encryption. In many cases, the gateway devices also act as a user interface, showing system status and notifications generated from the RPM system. Proprietary gateway devices exist, yet other smart devices can also be used, such as a user's smartphone or smart TV.
- Backend: The backend stores all the data collected, performs analysis, using data processing technologies to output reportes and alarms the user and caregiver if needed. It may also present an interface for medical professionals to explore the data and integrate with other systems.

### 2.2.2 System Categorization

Based on the proximity of computation to user, RPM system can be classified into the following categories:

- Cloud-based: The data analysis is performed in the cloud, which can leverage a large amount of computing resources for complex models. This is the most common model, and it minimizes the requirement to the user-side deployment, and consolides all the data under a central chokepoint, making it easier to secure and manage. However, the main issue is the limited bandwidth and unpredictable latency of transmitting real-time data streams across the Internet.

- Edge / Fog-based: These systems utilize the recent edge computing and fog computing trend, and brings the computation to nodes or facilitates much near to the user in the sense of networking distance. This drastically reduces latency, while also allowing relatively powerful models to be used for data analysis. Yet the separation between edge nodes and central backend further increases the system complexity, and the increasing number of nodes also poses new challenges in system management.

- Local-based: Using local devices for computation minimizes the latency, the risk of security issues and the amount of outward bandwidth required. Remote backend still exists, but only for sending alarms for emergency personalities and caregivers, or save data asynchronously for future analysis. However the computation capacity of local devices also constrains the model that can be used, therefore simple rule-based models that have strict response time requirements are more suitable then complicated deep learning models.

## 2.3 QoS in e-health

One major challenge for e-health systems is to provide Quality of Service (QoS) guarantee. Here the QoS requirements usually considers bandwidth, latency, error rate and priority. As an example, European Union's MobiHealth project [14] tried to build a mobile patient monitoring system, and experienced issues caused by delay variation and limited outbound bandwidth when using 3G network. The complexity of implementing QoS in these systems mainly comes from its heterogeneous nature, as multiple services each with different bandwidth and latency requirements may coexist. The service here may belong to different categories, such as tele-consultation (mainly Video/Audio chat, can tolerate data loss in exchange of latency) and tele-monitoring (mainly data collection, can not accept data loss), or belongs to the same category yet has different resolutions, such as a remote monitoring system performing simple 1-D blood pressure monitoring and multi-head Electrocardiogram (ECG) monitoring. Moreover, even for the same service, the QoS requirement may be specific to context, such as emergency versus non-emergency situations.

Many researchers have proposed solutions focusing on different aspects of e-health systems, attempting to provide better QoS support. In [15] the authors built a list of QoS requirement of common e-health services and mapped them to The 3rd Generation Partnership Project (3GPP)'s QoS Class Identifier (QCI) standard, providing a framework for developers and network operators to meet e-health service requirements. In [16], the authors presented a scheduling model that handles telemedicine traffic with higher priority while also fulfilling QoS requirements of other traffics, and in their simulation they are able to control the delay and loss below upper bound when the telemedicine traffic composes 10% of all traffic. In [17], the authors take the idea of context and propose 4 data transmission modes based on risk level and severity, from real-time continuous transmission to on-demand triggering, with the highest risk patients using more pressure. In [18], the author introduced "Bidirectional QoS control”, allowing the physician to manually reprioritize the data transmission to better suit the medical need, providing better flexibility when the network status worsens. In a similar idea, the author of [19] incorporated a set of tolerable degradations specified by the user, so network resources can be allocated from the user’s point-of-view to satisfy expected quality.

## 2.4 Sleep Stage Prediction

Using polysomnogram (PSG), the current gold standard, for sleep measurement requires a sleep technician to manually analyze sensor recordings and label the sleep stages, which is costly, time consuming and labor intensive, limiting its scalability. Therefore, many techniques have been developed to predict the sleep stages directly from raw sensor outputs during the PSG session. In [20], the authors proposed a multi-task 1D Convolutional Neural Network (CNN) network for automatic sleep staging using PSG sensor readings, which not only predicts the sleep stage of a single time slot but also predicts the stage of neighbouring slots, forcing the model to learn from contextual information. Their model is validated on 2 public datasets and yield an overall 5-class (Wake, Rapid Eye Movement (REM), N1, N2, N3) classification accuracy of 83%. In [21], the authors generates high resolution images from PSG signals, which then later used to train a deep and dense 2D CNN network inspired by advances in image classification tasks. From their experiments this architecture can extract better features, reaching an 93% accuracy for 5-class classification task.

To overcome PSG’s disadvantage of high setup complexity, researchers also tried to use other lower cost signal recording devices. In [22], a Bidirectional Long-Short Term Memory (LSTM) model is used for a 4-class (Wake, Light Sleep, Deep Sleep, REM) sleep stage prediction task using only wearable ECG as input, allowing for lower cost measurement while still being useful with an accuracy of 80%. In [23], the authors were able to perform real-time sleep stage classification using a wearable

ECG as data source and a time-distrbuted 1D CNN model running on a smartphone with 30-second epoch and 83.5% overall accuracy for the 5-class task.

## 2.5 Federated Learning

Federated Learning (FL) is a learning paradigm that trains on decentralized datasets for privacy preservation [24]. With Google's successful application of Federated Learning in next word prediction for mobile keywords [25] in 2018, it gradually gained popularity, stimulating discussions in system design [26], service abstraction [27], personalized experience [28], simulation [29] and many other aspects. The ability of learning from distributed datasets makes Federated Learning a good candidate not only for individual users' privacy protection, but also for bridging data silos. Another advantage of Federated Learning is its generalizability, allowing existing models to be used directly without significant structural changes. However, like any new technology, Federated Learning also introduced a new set of challenges [26], such as synchronizing training progress, steering training pace, managing participant population, dealing with data and device heterogeneity, and reducing communication cost.

Due to the sensitive nature of medical data, Federated Learning is especially suitable for medical applications. In [30], the author gave a comprehensive review of the usage of Federated Learning in Healthcare, including patient similarity learning, in-hospital mortality predicting and future re-hospitalization predicting. Additionally, other privacy-preserving technologies such as Differential Privacy can be used along with Federated Learning [31] to comply with stricter regulatory requirements.

## 2.6 Authentication

Authentication advancements can be classified into 2 categories: the introduction of new authentication factors, and exploration of combining multiple authentication factors. Out-of-band authentication is a subcategory of multi-factor authentication, which utilizes a separate communication channel for a secondary authentication factor different from the primary factor. Several out-of-band authentication methods are proposed in the literature including One-Time Password (OTP) based, mobile-based, and ID-based methods [32]. Novel factors such as acoustic signals[33], brainwave[34] and heartbeats (based on ECG) [35] are also being studied.

## 2.7 Molecular Communication

Though the concept of Molecular Communication (MC) in biological settings has been well studied for long, it is a relatively new idea in the networking and communication field. MC was first used as an engineering term in 2005 by T.Nakano, et.al [36], with the author proposing a signaling network for nanomachines working in an aqueous environment. Later there begins research on simulation, modulation analysis and error correction. The first "proof-of-concept" macro-scale experiment of molecular communication happened in 2013 [37], in which N. Farsad, et. al showed a modified spray filled with isopropyl alcohol used as a transmitter and alcohol sensors connected to an Arduino used as receiver, while fans were used to create artificial airflow in order to facilitate propagation. In 2017, N. Farsad, et.al further refined their decoding method using deep learning [38], with multiple architectures including Multi-Layer Perceptron (MLP), CNN and LSTM, and showed the effectiveness of their method in an aqueous setting experiment. In 2018, N. Farsad et.al proposed a new network architecture named Sliding Bidirectional Recurrent Neural Network (RNN) for continuous long sequence detection in an aqueous setting [39].

# Chapter 3

# Analyze: Application of Deep Learning and Federated Learning to Sleep Stage Prediction Task Using Data From Wearable Device

With the awakeness of privacy and data ownership, users trust less in centralized clusters and demand data collected to be kept on device. This creates new challenges for e-health application developers, who need to build distributed architectures to fulfill these expectations, without loss of performance. Deep Learning has been shown to be a powerful tool given large and diverse datasets, while Federated Learning showed great potential in decentralized training and privacy perversion, making both great candidates. In this chapter, we try to apply these two technologies to the sleep stage prediction task, using data collected using Apple Watch. We found that more advanced Deep Learning models can indeed outperform existing Machine Learning models, improving both prediction accuracy and stability. We also found that Federated Learning is able to achieve similar performance, though at the expanse of more data exchange and prolonged training time.

## 3.1 Introduction

Sleep is an essential activity that keeps the human body and minds functioning, yet it is often overlooked and neglected. Nowadays, many people suffer from sleep disorders, leading to an inadequate level of sleep in duration or quality. Nearly half of all Americans feel sleepy during the day between 3 and 7 days a week [40], and 35%

of all adults report sleeping on average less than 7 hours per night [41]. On one hand, research has been conducted to diagnose possible causes resulting in sleep disorders, including excessive consumption of stimulants such as caffeine, overexposure to blue light coming from smartphones and other devices, and high stress levels. On the other hand, sleep disorders have a profound negative effect, often associated with reduced productivity, inability to focus and even deterioration of health, both physically and mentally.

In order to improve one's sleep, the sleep disorders must first be observed and measured. Currently the gold standard for sleep measurement is polysomnogram (PSG). The patient comes to a dedicated sleep lab, with sensors monitoring multiple physiological parameters, such as brain wave, oxygen level, heart rate and eye movements. Later the recorded data is interpreted by a sleep technician to mark the stages of sleep during the sleep session and finally a sleep report is generated. PSG has been verified to be an effective and accurate technique, yet the costly and complicated process stopped it from reaching a broader audience. PSG is also restricted to hospital usage only, as it's not practical to install the full device in one's home, and the results from one PSG measurement can only represent the situation of one time point, instead of showing the whole trend. New methods need to be developed to obtain a long-term, continuous sleep measurement while keeping a low cost and complexity.

Thanks to the recent advance of consumer grade wearable devices, such as smartwatches and smart bands, they provide a new and ideal data source for sleep measurement. On the functionality aspect, most wearable devices already have an accelerometer on board for step counting purposes, while more devices start to include a heart rate sensor. On the user habit standpoint, users have been educated to accept wearing their wearable devices during sleep. Compared to PSG, sleep measurement using wearable devices is easier to use, more widely available while cost less, making it a tempting alternative. Many manufacturers have already exploited this idea, introducing proprietary sleep scoring systems, such as the Fitbit sleep score. However such systems are opaque, with their inner mechanisms either unknown or unvalidated to medical standards, restricting their usefulness.

In [4], the authors first developed their own application to acquire raw sensor data from Apple Watch, then conducted experiments having participants wearing Apple Watch during PSG sessions to obtain a dataset with sleep stage labels. Later the authors applied various Machine Learning algorithms to this dataset, trying to build a model that takes in raw sensor measurements and predicts the sleep stage. Finally they validated the model they obtained with other sleep datasets, which is indeed effective for a larger dataset containing more diverse samples called MESA [42]. Unlike proprietary scoring from device manufacturers, their work creates new possibilities for open and accurate sleep measurement using consumer grade wearable devices. However, the models in [4] are rather limited and simple, which process each sample point individually, ignoring the contextual information hidden in the time axis that could be extracted when considering neighbouring samples. Besides, the training approach used requires a centralized dataset, which may lead to privacy concerns on personal level and compliance issues on hospital level. Medical data are extremely sensitive, and even with anonymity procedures taken, hospitals may still be unable to combine their datasets to build a centralized dataset due to regulatory requirements.

In this section, based on the work in [4], we tried to tackle both the model problem and the privacy problem. Specially, we attempted to answer the following research questions:

- RS1. (Performance) Can Deep Learning and more advanced network structures work better for the sleep stage prediction problem?
- RS2. (Privacy & Data Security) Can the federated learning concept be applied to the stage prediction problem to achieve better privacy and data security

guarantees?

On the performance side, to improve the accuracy of the sleep stage prediction task, the task is reframed as a dense-label time prediction problem and more advanced and complicated deep learning models are utilized. On the privacy side, we try to apply the federlated learning paradigm for this task, which enables training useful models from decentralized datasets. From experiment results, the accuracy and consistency of the sleep stage prediction both increased when more suitable models are used, and federated learning produced models that's comparable to models trained using a centralized dataset.

## 3.2 Problem Description

### 3.2.1 Dataset

The dataset used in this thesis is the same dataset used in [4].[1] To create the dataset, 39 health subjects were given an Apple Watch and wore it for a week, recording their activity pattern to learn about their circadian clock. At the end of the week, they took a PSG measurement while wearing an Apple Watch. A custom application is developed to read raw acceleration and heart rate data from Apple Watch's sensor and send it to a server wirelessly. To be compatible with other existing ECG dataset such as MESA, 3-axis acceleration readings are converted into activity counts. After the PSG session, technicians label the sleep stage for each time point (epoch), and data from Apple Watch are aligned to the labels. In the final inspection, data from 8 subjects are excluded due to medical conditions or data logging issues.

The final dataset contains 31 sleep episodes, one-to-one associated to 31 subjects, with an average episode length of 821. Each episode contains multiple samples. Each sample represents a 30 second window, including 1 label and 4 features. The label

[1]Motion and heart rate from a wrist-worn wearable and labeled sleep from polysomnography v1.0.0. Available at `https://physionet.org/content/sleep-accel/1.0.0/`

Table 3.1: Classification Task Simplification

| Sleep Stage | Wake | REM | N1 | N2 | N3 |
|---|---|---|---|---|---|
| 2 Class | Wake | Sleep | | | |
| 3 Class | Wake | REM | NREM | | |

is annotated by the technician from PSG recordings, and falls into one of 5 possible sleep stages: Wake, REM, N1, N2 and N3. Each feature is a single value, and they are generated as follows:

- Activity count, converted from accelerator readings
- Heart rate feature, smoothed using Gaussian filters and aggregated by computing the standard deviation
- Proxy of circadian clock, using a fixed cosine wave calculated from time for easy computation
- Proxy of circadian clock, built from a well-validated mathematical model utilizing activity patterns recorded before

### 3.2.2 Task

Although the label for each sample (window) has 5 possible classes, the data from wearable devices is not sufficient enough to give a good accuracy on the 5-class classification task. Therefore, in [4], the authors lowers the difficulty of the sleep stage prediction task by aggregating some classes into one big class. Specifically, all the classes beside Wake can be aggregated into a single class called **Sleep**, and all the classes beside Wake and REM can be aggregated into a single class called **NREM**, as shown in Table 3.1.

Reducing the number of possible classes makes the model more likely to give out a correct prediction while remaining useful for long-term continuous monitoring purposes.

Following the task definition in [4], the sleep stage prediction task can be divided into 2 subtasks:

- (SW) Classify each sample into 2 classes, Sleep or Wake
- (Three) Classify each sample into 3 classes, Wake, REM or NREM

### 3.2.3 Formulation

A formal formulation of the task can be represented as follows. Dataset $D$ contains $n$ episodes, and $E$ represents an episode, i.e $D = \{E_1, E_2, \dots E_n\}$. Each episode $E$ contains $t$ samples, which are placed according to time, from earliest to latest, i.e $E_i = S_0, S_1, S_2, \dots S_{t-1}$, with subscript indicating the time step. Each sample $S$ further contains a feature vector $X$ of $d$ dimension and a class label $Y$, i.e $S_j = (X_j, Y_j)$. The goal is to train a model that takes in an episode $E_{in}$, and output a predicted label for each sample (each time step) in this episode, resulting in a predicted class vector $Y_{out}$ having the same length as $E_{in}$.

## 3.3 Models

Many deep learning models have been developed to solve classification problems. For this specific problem, applicable models can be classified into 3 categories, each having different input and output shapes, as illustrated in Fig 3.1 and detailed below:

- One-to-one: Models in this category take 1 sample as input, and return 1 predicted class label as output. Each sample is considered individually, without any contextual information from neighbouring samples. Since each sample represents a 30 second window, the model can only use the information contained in this small window. This is the Neural Network model used in [4], also known as Multi-Level Perceptron (MLP) in Deep Learning contexts. However, in this thesis, the MLP structure used is different, as shown in Table 3.3.

Table 3.2: Model Categories

$w$ stands for window size, $s$ stands for step size.

| Model Category | Input & Output | Parameter | Model Names |
|---|---|---|---|
| One-to-one | $1 \times d \rightarrow 1$ | None | MLP |
| Many-to-one | $w \times d \rightarrow 1$ | $w$ | CNN, LSTM, BiLSTM, CNN-LSTM, CNN-BiLSTM |
| Many-to-many | $w \times d \rightarrow w$ | $w$, $s$ | LSTM, UNet |

- Many-to-one: Models in this category take a window of $w$ samples as input, and return 1 predicted class label, which represents the label for the sample in the middle of the sliding window. Every sample in the input episode has its own window. For the samples on the boundary of an episode, the boundary sample is repeated to pad the window. Introducing sliding windows allows the model to utilize contextual information, but also results in additional computational costs, since the network is more complex and has a larger input. CNN, LSTM, BiLSTM and CNN-LSTM are models used in this thesis that fall in this category.

- Many-to-many: Models in this category take a window of $w$ samples as input, and return $w$ predicted class labels, representing the label of each sample in the input window. For next prediction, the window move forward in time with a step (stride) size of $s$. Boundary samples are again padded using itself. Finally, after the window reaches the end of the sequence and the whole episode has been processed, each sample may have multiple labels (as windows are overlapped), and these labels are aggregated by average to get the final label for that sample. LSTM and UNet [43] are models used in this thesis that fall in this category.

Also, thanks to the generalizability of federated learning, all the models have 2 variants, one trained using a centralized dataset, the other one trained using decentralized dataset, in which each client contains an episode.

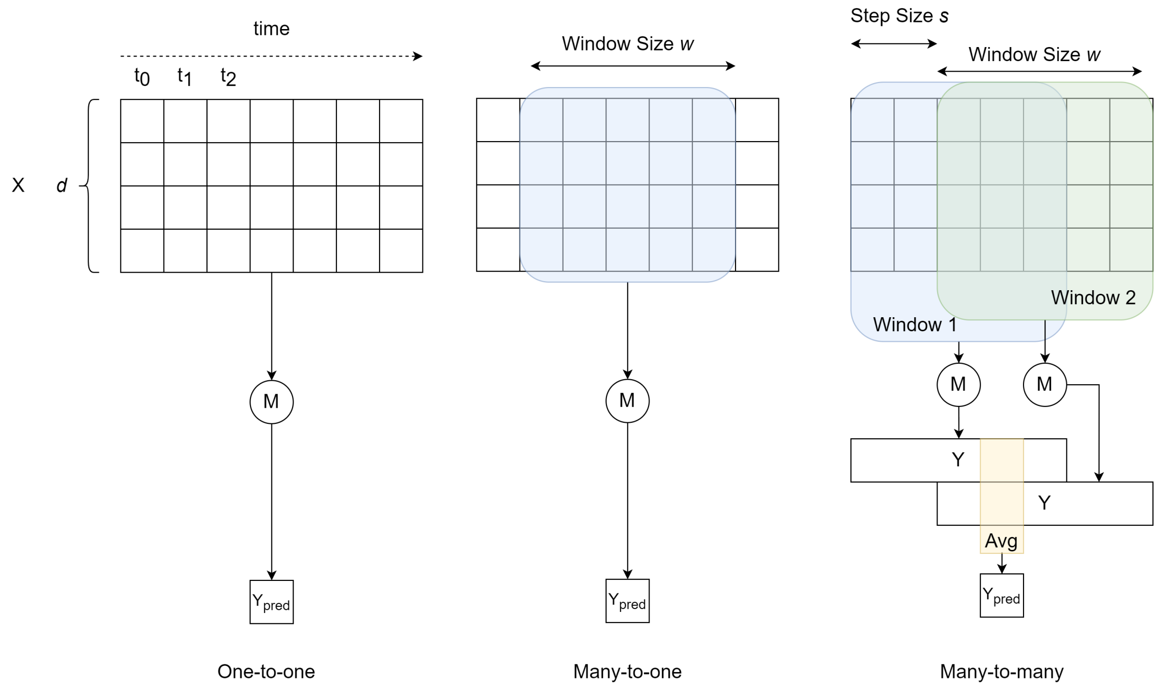

Figure 3.1: Visualization of Model Categories

Table 3.3: Model Structure

| Model Category | Model Name | Structure |
|---|---|---|
| One-to-one | mlp | dense(128), dense(32), dropout(0.3), dense(16) |
| | (baseline) | dense(15), dense(15), dense(15) |
| Many-to-one | cnn | conv1d(3), conv1d(3), dropout(0.3), maxpool1d(2), flattern(), dense(64) |
| | lstm_one | lstm(64), dropout(0.3), dense(64) |
| | lstm_one_bi | bilstm(64), dropout(0.3), dense(64) |
| | cnn_with_lstm | cnn: conv1d(3), conv1d(3), maxpool1d(2), flattern()<br>lstm: lstm(32), lstm(32), flattern()<br>backend: concat(cnn, lstm), dropout(0.3), dense(64) |
| | cnn_with_lstm_bi | cnn: conv1d(3), conv1d(3), maxpool1d(2), flattern()<br>bilstm: bilstm(32), bilstm(32), flattern()<br>backend: concat(cnn, bilstm), dropout(0.3), dense(64) |
| Many-to-many | lstm | lstm(64), dropout(0.3), dense(64) |
| | unet | unet1d(depth=3, width=32, kernel=3, channel=4) |

## 3.4 Experiment

**Dataset Split** Following the experiment design in [4], in all experiments, 70% of the episodes are assigned as training set and the remaining 30% are assigned as test set. For federated learning, to simulate a possible scenario of participants keeping their own data locally, the training set is split into multiple clients, with each client only having 1 episode. The testing set in federated learning is implemented as centralized, but it would be equivalent to having each test client containing their own episode locally.

**Parameters** Compared to federated learning, centralized learning is both faster and easier to implement, therefore most parameter tuning decisions are made on centralized training experiments, and later applied to the federated learning experiment. The detailed parameter ranges are listed as below:

- For many-to-one models trained with a centralized dataset, window sizes are chosen from 11 to 101 (stepping 6).
- For many-to-many models trained with a centralized dataset, window sizes are chosen form 11 to 101, (stepping 6); step sizes are chosen from 5 to 25 (stepping 5).
- For federated learning, after analyzing the results of centralized training, window size is set to 32 for many-to-one models and additionally step size is set to 8 for many-to-many models.

**Training** For centralized training, 20% of the training set is chosen as validation set. Adaptive Moment Estimation (Adam) optimizer with 1e-3 learning rate is used, along with a learning rate reduction strategy monitoring the loss on the validation set, reducing learning rate to a tenth when reaching plateau. Models are allowed to train as much as 100 epochs, but an early stopping mechanism stops the training if the loss on validation set begins to increase for 5 epochs. For federated learning, following the advice on [44], FedAvg is used for aggregation, with both server and client using Stochastic Gradient Descent (SGD) optimizer, but server using a learning rate of 1.0 and client using a learning rate of 0.02 to prevent overfitting. Models are all trained for 20 rounds, where 1 round is approximately equivalent to 5 epochs (set by the client data fetching strategy) in centralized training.

**Evaluation** All models are evaluated using the same mechanism as in [4] to enable direct comparison. To minimize the perturbation caused by randomness, Monte Carlo cross-validation is used. All centralized models are trained 10 times, each time with a different training and testing split. Unless stated otherwise, all metrics values reported are average values aggregated along all these repeated experiments. The important metrics are explained as follows, with more reasoning can be find in [4]:

- (SW) Area Under Curve (AUC): The area under the Precision-Recall curve, with recall (x-axis) is the fraction of wake predicted correctly and precision (y-axis) is the fraction of predicated wake being labelled wake in ground truth. AUC is chosen here due to the class imbalance.

- (Three) Best Accuracy: The best classification accuracy found when searching through the threshold space composing 2 thresholds. The first threshold is used to predict whether the label is wake or not, and the second threshold decides whether the label is REM or NREM.

- (Three) Kappa: Cohen's kappa coefficient between the predicted labels and ground truth. Kappa measures the inter-rater reliability of categorical predictions, and takes the possibility of reaching agreement by chance. It is more robust than simple agreement percentage calculation and is widely used in medical researches to measure the agreement of diagnoses using different methods.

Table 3.4: Top 10 Models

Ordered using the summation of improvement on (SW) AUC and (Three) Best Accuracy.

| Category | Model Name | Window Size | Step Size | (SW) AUC | (Three) Best Accuracy | (Three) Kappa |
|---|---|---|---|---|---|---|
| Many-to-many | lstm | 12 | 5 | 0.899 | 0.744 | 0.369 |
| Many-to-many | lstm | 24 | 5 | 0.896 | 0.745 | 0.386 |
| Many-to-one | cnn_with_lstm | 11 | | 0.893 | 0.736 | 0.315 |
| Many-to-one | cnn | 23 | | 0.886 | 0.751 | 0.401 |
| Many-to-one | lstm_one_bi | 35 | | 0.883 | 0.741 | 0.407 |
| Many-to-many | unet | 24 | 5 | 0.883 | 0.745 | 0.395 |
| Many-to-many | lstm | 54 | 5 | 0.883 | 0.759 | 0.422 |
| One-to-one | mlp | | | 0.881 | 0.724 | 0.273 |
| Many-to-one | cnn_with_lstm | 23 | | 0.878 | 0.744 | 0.386 |
| Many-to-one | cnn | 47 | | 0.878 | 0.757 | 0.418 |
| Baseline [4] | baseline | | | 0.878 | 0.723 | 0.277 |

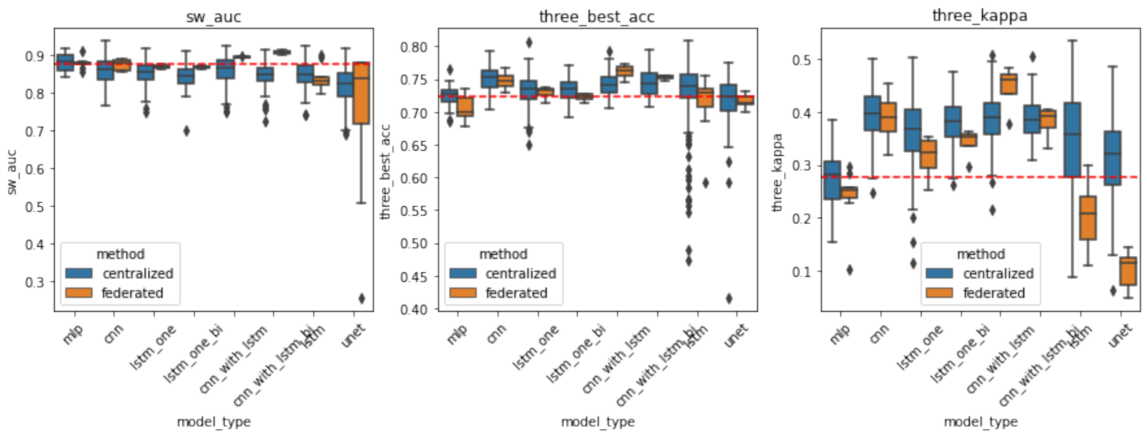

Figure 3.2: Model Performance of Centralized Training and Federated Training

## 3.5 Discussion

**Usefulness of Deep Learning Models** The best results in [4], obtained using a Neural Network, are used as a baseline, displayed as a red dash in related figures. For the sleep/wake classification task, the introduction of deep learning doesn't provide much benefit, with only 7% of all deep learning models experimented can exceed the baseline accuracy. One possible explanation is that the distance between sleep and

Table 3.5: Data Transmission and Time Cost of Federated Learning

| Model Category | Model Name | Data Transmited Per Round (One Way, MB) | Time Per Round (Federated, Second) | Time Per Round (Centralized, Second) | Time Ratio (Federated/Centralized) |
|---|---|---|---|---|---|
| One-to-one | mlp | 3.4 | 28.7 | 0.2 | 143.6 |
| Many-to-one | cnn | 20.8 | 41.8 | 0.5 | 83.6 |
| Many-to-one | lstm_one | 14.1 | 403.2 | 1.2 | 336.0 |
| Many-to-one | lstm_one_bi | 28.0 | 743.0 | 1.3 | 571.6 |
| Many-to-one | cnn_with_lstm | 30.4 | 809.8 | 3.5 | 231.4 |
| Many-to-one | cnn_with_lstm_bi | 45.4 | 1610.4 | 4.2 | 383.4 |
| Many-to-many | lstm | 14.1 | 93.7 | 3.1 | 30.2 |
| Many-to-many | unet | 432.6 | 120.9 | 6.1 | 19.8 |

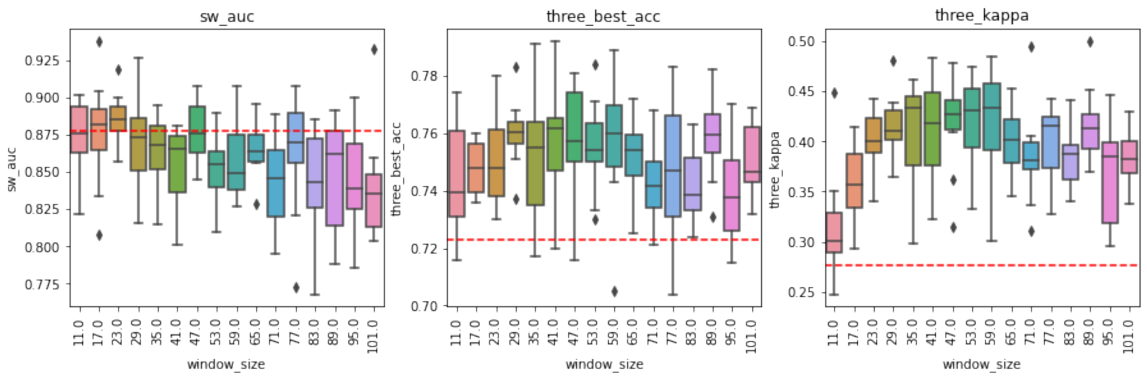

Figure 3.3: Effect of different window size on many-to-one CNN model

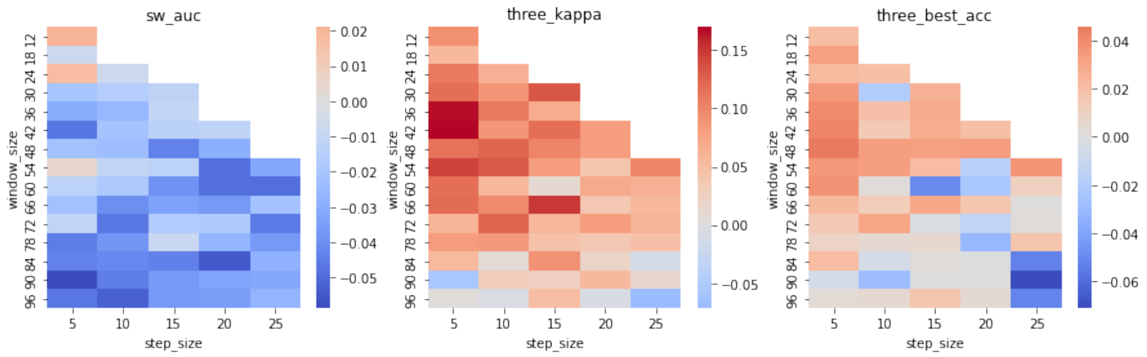

Figure 3.4: Effect of different window size and step size on many-to-many LSTM model

Baseline is marked as 0. Red indicates improvement and blue indicates deterioration.

wake is obvious and can be recognized easily, and using a complex model for such a simple problem will lead to overfitting. The best model improved 2.1% on the 2-class classification accuracy, which is a many-to-many model using a classic LSTM structure, known for its ability to capture contextual information. For the sleep/N-REM/REM classification task, deep learning models consistently perform better than the baseline, with 79% of the models experimented achieving a better best accuracy and 92% having a higher kappa value. As the difficulty of the task increases, the value of using a more capable model begins to be recognized, which allows the model to find out the subtle differences between different sleep stages. The best model improved 4.5% on the 3-class classification accuracy, showing again the power of LSTM and the importance of context.

**Effect of Window Size and Step Size** The sliding window technique provides a simple and effective way for models to take in contextual information, at the cost of additional computation. In Fig. 3.3, the changes of key metrics with the window size is shown, using many-to-one CNN models as example. For the sleep/wake classification task, a small window already allows the model to perform well, and the accuracy suffers when the window keeps increasing. However, for the harder sleep/N-REM/REM classification task, a larger window indeed provides a boost to the model, improving both in the best accuracy and also the prediction consistency. Turning to many-to-many models, allowing the model to jump forward reduces the amount of computation, yet a sample can still benefit from multiple predictions by taking average of all the predicted probability values. In the case of LSTM models, a smaller step size seems to be beneficial, allowing one sample to have more overlapping predicted labels, while larger step size hurts performance, as shown in Fig. 3.4.

**Exploration of Federated Learning** Compared to their centralized counterparts, the models trained using federated learning generally perform slightly worse, generally around 2%. That is understandable, since in federated learning setting models do not have full access to all the data. However, in most cases, the degradation of performance is not very significant. More surprisingly, in the sleep/wake classification task, some many-to-one models even saw better accuracy using federated learning instead of centralized learning. A possible explanation is that the difficulty caused by the limitation of local data only may have a regularization effect, forcing the model to squeeze out more from the data it has access to. Beside performance issues, federated learning also introduced additional communication cost, since it has to constantly exchange parameters to enable the training to progress. Waiting for all clients to synchronize also introduces extra time cost in each round, and more time is needed with the number of clients increasing. In centralized training, a round may only take 2s, but in distributed training the time is usually amplified by 10 to 500 times, as shown in Table 3.5. But after all, if the time and network cost is acceptable (for example, when the training happens during night, and the model is not immediately required), Federated Learning is a viable solution once an efficient model is chosen, and possibly the only solution when data security and privacy is taken into account.

This chapter is built on the works done in [4], and tries to improve on both the model performance and privacy protection. On the performance side, our experiments demonstrated that deep learning models show diminishing returns for the simple 2-class classification problem, with at most 2% improvement on accuracy. However the additional memory and contextual capabilities of deep models indeed paid off in the 3-class classification problem, achieving 4% improvement on accuracy and huge increase in consistency. On the privacy and data security side, federated learning provides a viable solution to the sleep stage prediction problem, protecting privacy at the cost of additional time, communication and slightly worsened performance. Still, there are many problems waiting to be explored. Models can be further refined, adding more layers and exploiting the recent progress of time series data research. Training can be more thoroughly tuned, with different optimizers and training strategies.

Non-iid data also presents new challenges to federated learning, which may happen if the participants have special and uncommon conditions. This work is only a preliminary step, and with further investigation there will arise better solutions to not only this sleep stage prediction task, but also other important and seemingly unsolved problems.

# Chapter 4

# Acquire: Delay Estimation in E-health Sensor Networks

Still, not all e-health applications can utilize a distributed architecture, and for some use cases, such as Remote Patient Monitoring systems, a central backend is almost a necessity. In such cases, the e-health system can still be improved by providing a better model for estimating the delay of data collection, defined as the time between the data generation and its arrival at the backend server. Limiting the delay to a certain range can help the system satisfy specific Quality of Service (QoS) requirements, which has long been a problem. In this section we analyzed a Wireless Sensor Network (WSN) under medical setting using both queueing theory and simulation, first obtained an analytical solution for the waiting time of first node, then examined the "catch up" phenomenon in latter nodes, and finally proposed a delay estimation using a small number of nodes to predict the estimation for large number of nodes. Experiments under different conditions show our estimation works reasonably well.

## 4.1 Introduction

Wireless Sensor Network (WSN) has seen wide applications in various fields, including manufacturing, automation, and medical monitoring, due to its ability to gather information from a vast distributed array of sensor nodes, sometimes even across geographical boundaries. Beside measuring the environment and obtaining readings, these sensor nodes can also perform computation, such as run distilled machine learning models to intelligently filter out unimportant readings, in order to reduce

Table 4.1: QoS Requirement of Common e-Health Applications

| Service | Example Use Case | Delay | Loss |
|---|---|---|---|
| Text chat[45] | Tele-consultation | 2-5 s | 0 |
| Audio chat[15] | Tele-consultation, Collaborative diagnosis | <150ms | <1% |
| Video chat[15] | Tele-consultation, Collaborative diagnosis | <250ms | <1% |
| Robotic service[46] | Remote surgery | <40ms | 0 |
| Sensor logging[15] | Tele-monitoring | <300ms | 0 |
| Automated alert[15] | Elderly care | <10s | 0 |

communication cost. The network topology for WSNs are often star-like, with one centralized server acting as the sink in the center, and all the sensor nodes placed in the network edge as the source.

In all the applications of WSNs, some require the network to be compliant to Quality of Service (QoS) requirements, especially in the medical field, such as remote monitoring for elderly care. QoS requirements commonly consist of 2 aspects: delay and packet loss, the former limits the maximum time for a packet to be transmitted to the sink from a source, and the latter specifies that the tolerable possibility of a packet being lost in transit, which may be caused by interference or buffer overflow. Table 4.1 shows some common e-health applications and their corresponding QoS requirements.

However, calculating the delay of a packet experienced in a WSN is not straightforward. It is possible to model WSN as a queuing system, and try to approach the problem using queuing theory, modeling the delay as the sum of response time across all nodes in the path from sink to source. However, as in many cases packet arrival is not strictly a Poisson process and the service time distribution is also not exponential, resulting in a G/G/1 queue at one intermediate node, which is known to have no analytical solution and can be only approximated. Even worse, beyond a single node, the interactions between different kinds of packets in a series of nodes are more complicated, adding more complexity.

In this thesis, we try to analyze a WSN spanning multiple hops intended for e-health monitoring applications via a simulation-based approach. We first show that although the behavior of packets in the first intermediate node can be analytically determined, in succeeding nodes the waiting time for low priority data packets gradually increases, invalidating the naive method of estimating the end-to-end delay using only the first node. Next, we discuss the "catchup" phenomenon, which partially explains the increase of waiting time for low priority packets, and provides a slightly better lower bound estimation for the end-to-end delay. Finally, we try to predict the delay over a long series of nodes, using only the simulation results of a short series of nodes, attempting to reduce the simulation time while keeping the results relatively close to the true values.

## 4.2 Problem Description

### 4.2.1 Application

The e-health application for this thesis focuses on elderly care, in which various sensors are installed throughout the home of elders, such as movement sensors for falling detection, gas sensors for preventing gas leak and relay nodes to forward the readings of vital signs obtained from smart bands. Fig 4.1 illustrates such a system. All the sensors send their data to the monitoring center, in which the data is archived and analyzed. Beside sensors, elders can also manually send messages to the monitoring center using a control panel, such as in the case of requesting help. There are 2 types of packets in the system, data and control, between which control has higher priority. Normally, the packets generated by the sensor nodes are data packets, and the messages sent by the user manually are control packets. However, sensors can mark a data packet as an emergency packet when there is an emergency situation, and emergency packets will have the same high priority as control packets. Priority in the system is non-preemptive, meaning a high priority packet can not interrupt an ongoing low priority packet that is already in service. The QoS requirement of the whole

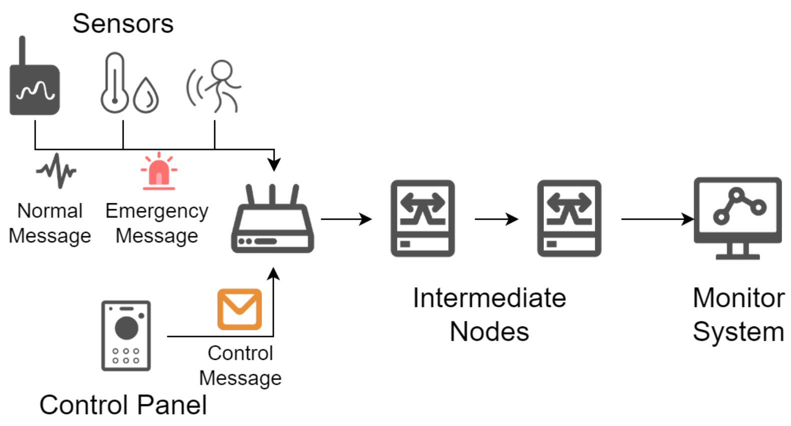

Figure 4.1: Use case in elderly care

system is on the data packets, whose end-to-end delay can not exceed a predefined threshold. The goal here is to determine the maximum number of intermediate nodes (hops) given the constraint.

### 4.2.2 Queueing System Formulation

The network model discussed consists of multiple sensor nodes at one end, and a sink at the other end, linked by a series of intermediate nodes. There are $N_{sensor}$ sensors, and each sensor node generates data packets following a Poisson process of arrival rate $\lambda_{data}$. Then the data packet containing readings are analyzed in the node using some onboard models, to decide the urgency of the data. This sensor analysis process has an exponential service time distribution with a service rate of $\mu_{data\_sensor}$. Every data packet has a probability of $P_H$ to be marked as an emergency packet. After the sensor finishes analysis, the packet is sent to the first intermediate node, which aggregates all the packets from all the sensors, as well as control packets from the control panel. Control packets are generated following a Poisson process of arrival rate $\lambda_{control}$, which directly goes to the first intermediate node. In all intermediate nodes, service time follows exponential distribution, and data packets are serviced with a service rate of $\mu_{data\_node}$ while control packets are serviced with a service rate of $\mu_{control\_node}$. After $N_{node}$ intermediate nodes, packets arrive at the sink and are

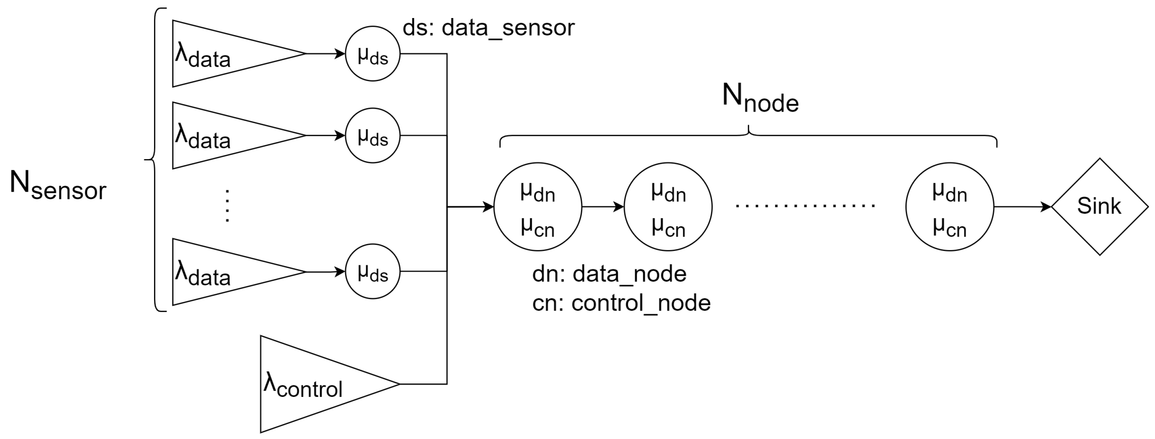

Figure 4.2: Abstracted System Model

Table 4.2: Model Parameters

| Notation | Meaning | Default Value |
|---|---|---|
| $N_{sensor}$ | Number of sensors | 4 |
| $N_{node}$ | Number of intermediate nodes | 10 |
| $\lambda_{data}$ | Arrival rate of data packets at each sensor | 2 |
| $\lambda_{control}$ | Arrival rate of control packets at the first intermediate node | 1 |
| $\mu_{data\_sensor}$ | Service rate of data packets at each sensor | 5 |
| $\mu_{data\_node}$ | Service rate of data packets at each intermediate node | 40 |
| $\mu_{control\_node}$ | Service rate of control packets at each intermediate node | 10 |
| $P_H$ | Probability of a data packet being marked as emergency | 0.2 |

removed from the system. The end-to-end delay of a data packet is modelled as the sum of response time of the data packet across all nodes, including the origin sensor and intermediate nodes. An illustration of the network is shown in Fig. 4.2. The notations, along with the default parameters used in the simulations, are listed in Table 4.2.

For simplicity, certain assumptions are made. First, there is no packet loss, and all the buffers are assumed to be infinite. Second, the propagation delay $\tau$ are outlined in certain formulas, but they are assumed to be negligible.

### 4.2.3 Response Time of Data Packet

We can derive the formula that expresses the response time of a data packet when there is $N_{node}$ intermediate nodes,

$$T_{resp\_data}^{N_{node}} = T_{resp\_data\_sensor} + \sum_{i=1}^{N_{node}} (T_{wait\_data\_i} + T_{srv\_data\_i}) \tag{4.1}$$

For the sensors, as there is only data packets, it can be easily formulated as an M/M/1 queue, which has the following analytic results:

$$T_{resp\_data\_sensor} = \frac{1}{\mu_{data\_sensor} - \lambda_{data}} \tag{4.2}$$

Also, for data packets, the service time at intermediate nodes is trivial to solve, just use the definition of service rate

$$T_{srv\_data\_i} = \frac{1}{\mu_{data\_node}} \tag{4.3}$$

Yet the waiting time of packets in intermediate nodes is problematic and could not be easily formulated, which will be detailed in later sections. Once the waiting time is obtained, the maximum hop count $N_{node}^*$ satisfying QoS limit for data packet $T_{QoS}$ can be trivially calculated by finding the integer that satisfies the following inequality:

$$T_{resp\_data}^{N_{node}^*} + \tau \leq T_{QoS} \leq T_{resp\_data}^{N_{node}^*+1} + \tau \tag{4.4}$$

## 4.3 Key Observations from Simulations

To get some intuition of the network's behavior, we performed various simulations, each with different parameter configurations. Here, we mainly focus on the waiting time of data packets at a certain intermediate node, indicated by the node index, which starts from 0 for the first intermediate node. Representative results are shown in Fig. 4.3 and Fig. 4.4. From those simulation results, the following 2 key observations were made.

**Key Observation 1.** As a data packet progresses into the network and arrives at deeper nodes (which have higher node index), the waiting time of the data packet

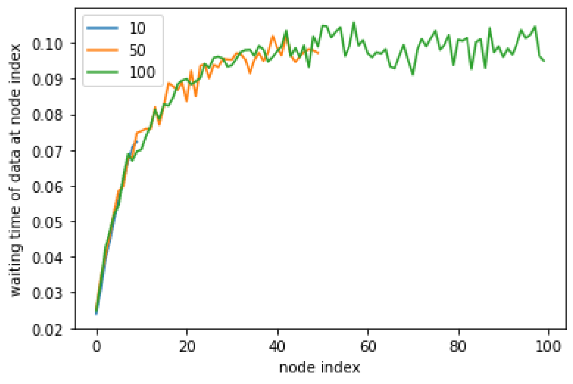

Figure 4.3: Average waiting time of data packet at intermediate node of given index, with $N_{node}$ of 10, 50 and 100

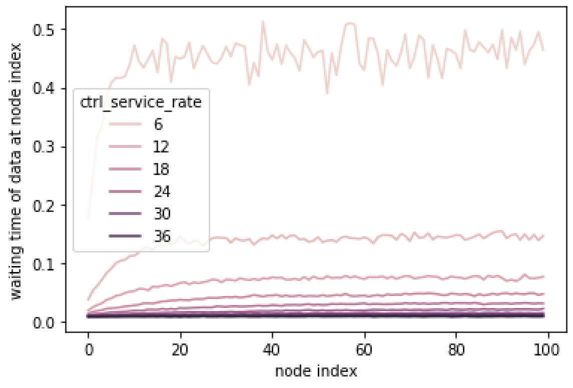

Figure 4.4: Average waiting time of data packet at intermediate node of given index, with $\mu_{control\_node}$ from 6 to 36

at the node increases.

**Key Observation 2.** If the waiting time at deeper nodes is able to reach an equilibrium, adding more nodes to the network will not cause further increase of waiting time at intermediate nodes.

## 4.4 Modeling the First Intermediate Node

We first try to characterize the behavior of packets at the first intermediate node, which is the aggregation node of all data, emergency data and control packets. As the service rate of data and control packets are different at intermediate nodes, it can not be simply modeled as an M/M/1 queue, but an M/G/1 queue.

However, we can still combine the control packets and emergency data packets together to form a "high priority" type of packets, and assign the (normal) data packers as a "low priority" type of packets. This trick allows the first node to be modeled as a special case of M/G/1 queue, namely M/M/1 queue with non-primitive priority. The formulation is as follows.

First we calculate the first and second order moment of service time of both priorities. As the Poisson process can be combined, the total arrival rate of data packets can be seen as the sum of the arrival rate of all the sensors. For an exponential distribution with mean $1/\mu$, the second order moment is $2/\mu^2$.

$$E[S] = \frac{N_{sensor}\lambda_{data}}{N_{sensor}\lambda_{data} + \lambda_{control}} \frac{1}{\mu_{data\_node}} + \frac{\lambda_{control}}{N_{sensor}\lambda_{data} + \lambda_{control}} \frac{1}{\mu_{control\_node}} \tag{4.5}$$

$$E[S^2] = \frac{N_{sensor}\lambda_{data}}{N_{sensor}\lambda_{data} + \lambda_{control}} \frac{2}{{\mu_{data\_node}}^2} + \frac{\lambda_{control}}{N_{sensor}\lambda_{data} + \lambda_{control}} \frac{2}{{\mu_{control\_node}}^2} \tag{4.6}$$

Then we can calculate the excess $E[S_e]$, which is the remaining service time of the packet in service as seen by an arrival packet.

$$E[S_e] = \frac{E[S^2]}{2E[S]} \tag{4.7}$$

Next, we calculate the utilization of high priority queue $\rho_H$, low priority queue $\rho_L$ and the whole queue $\rho$, using the definition of utilization $\rho = \lambda/\mu$. Note here the emergency data packets are having high priority, therefore it is counted in the utilization of $\rho_H$.

$$\rho_H = \frac{\lambda_{control}}{\mu_{control\_node}} + \frac{\lambda_{data} P_H}{\mu_{data\_node}} \tag{4.8}$$

$$\rho_L = \frac{\lambda_{data}(1 - P_H)}{\mu_{data\_node}} \tag{4.9}$$

$$\rho = \rho_H + \rho_L \tag{4.10}$$

Finally, we can apply the formulas of M/M/1 with non-preemptive priority [47] to obtain the waiting time of high priority queue and low priority queue.

$$E[T_Q^H] = \frac{\rho E[S_e]}{1 - \rho_H} \tag{4.11}$$

$$E[T_Q^L] = \frac{\rho E[S_e]}{(1 - \rho_H)(1 - \rho_H - \rho_L)} \tag{4.12}$$

And the analytic results of the first node indeed agree with the simulation results, as shown in Fig. 4.5.

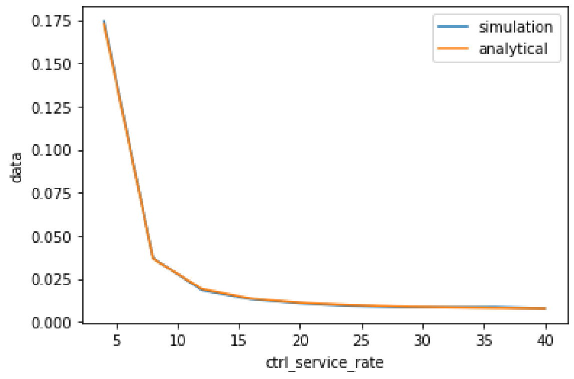

Figure 4.5: Average waiting time of data packets in the first intermediate node, with $\mu_{control\_node}$ from 4 to 40, obtained using analytic solution and simulation

## 4.5 Catchup Phenomenon in Subsequent Nodes

One possible explanation of the increase of data packets' waiting time at deeper nodes is the "Catchup" phenomenon, which says that overall, more and more high priority packets will be queued before data packets when they arrive at deeper nodes.

The intuition is as follows: Although physically for each intermediate node, there is only one queue and all the packets get into that queue, mentally we can separate the single queue into 2 queues, queue high ($Q_H$) that only contains high priority packets (i.e control and emergency data) and queue low ($Q_L$) that only contains low priority packets (i.e data). For a low priority data packet $D$, when it arrives at an intermediate node $N_i$ and waits to be serviced, some high priority packets will arrive later but serviced earlier during the waiting. After the data packet $D$ has passed the node $N_i$ and arrives at the next intermediate node $N_{i+1}$, the high priority packets that suppressed the data packet in the previous node will be already present in the queue high of current node $N_{i+1}$, therefore introducing additional waiting for that data packet at current node. In other words, some high priority packets catch up from behind and suppress the low priority packets, as shown in Fig. 4.6.

To formalize the effect of catchup phenomenon in intermediate nodes subsequent to the first node, we tag a data packet $D$ at node $N_i$. Assuming this packet's waiting

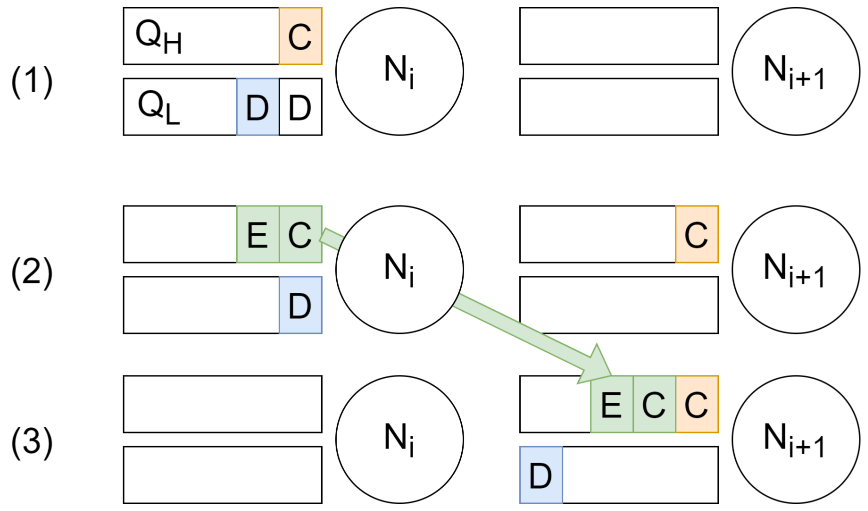

Figure 4.6: The catchup phenomenon

At time 1, our tagged data packet $D$ (in blue) arrives at node $N_i$'s $Q_L$. At that time, there is some data packet ahead in $Q_L$, and some high priority packet in $Q_H$ (in yellow). At time 2, when our tagged data packet is still waiting to be serviced, some new high priority packets (in green) arrive at $Q_H$, and will be served before $D$. At time 3, $D$ has just been served at node $N_i$ and arrives at the next node $N_{i+1}$, here it sees the high priority packets that catched up already in $Q_H$.

time at node $N_i$ is $t_i$ and the waiting time at node $N_{i+1}$ is $t_{i+1}$. Here to simplify the formulas, we let $\mu_d = \mu_{data\_sensor}$, $\mu_c = \lambda_{mu\_sensor}$, $\lambda_d = \lambda_{data} N_{sensor}$, $\lambda_c = \lambda_{control}$.

For the data packet $D$, during its waiting at $N_i$, there will be $N_c$ control packets and $N_e$ emergency data packets catch up.

$$N_c = \lambda_c t_i \tag{4.13}$$

$$N_e = \lambda_d P_H t_i \tag{4.14}$$

And these packets will result in an overhead (time) of $t_c$ for control packets and $t_e$ for emergency data packets, which will be $t$ in total.

$$t_c = \frac{N_c}{\mu_c} = t_i \frac{\lambda_c}{\mu_c} \tag{4.15}$$

$$t_e = \frac{N_e}{\mu_d} = t_i P_H \frac{\lambda_d}{\mu_d} \tag{4.16}$$

$$t = t_c + t_e = t_i \rho_1 \tag{4.17}$$

That means that at the next node $N_{i+1}$, the data packet $D$ need to wait for additional time $t$ beside the packets that's originally in node $N_{i+1}$ when $D$ arrives at node $N_i$. Assuming that when the system becomes stable, the number of high priority packets in high queue should remain relatively the same across all intermediate nodes, meaning that the waiting time for high priority packets will be $E[T_Q^H]$ before any catchup happened. Therefore, the waiting time at next node $t_{i+1}$ can be formulated as

$$t_{i+1}(1 - \rho) = \rho E[S] + \rho_1(E[T_Q^H] + t_i) \tag{4.18}$$

Which after some solving, gives

$$t_{i+1} = \frac{\rho E[S] + \rho_1(E[T_Q^H] + t_i)}{(1 - \rho)} \tag{4.19}$$

Now we try to apply this formulation to obtain an updated approximation of the waiting time, setting $E[T_Q^H]$ as $t_0$ and solve recursively. From Fig. 4.7, it can be seen that this approximation is slightly better than the naive approximation only using the first node, yet it is still far from perfect. The problem of this approach is that it converges too fast, much faster than the waiting time curve derived from simulation results.

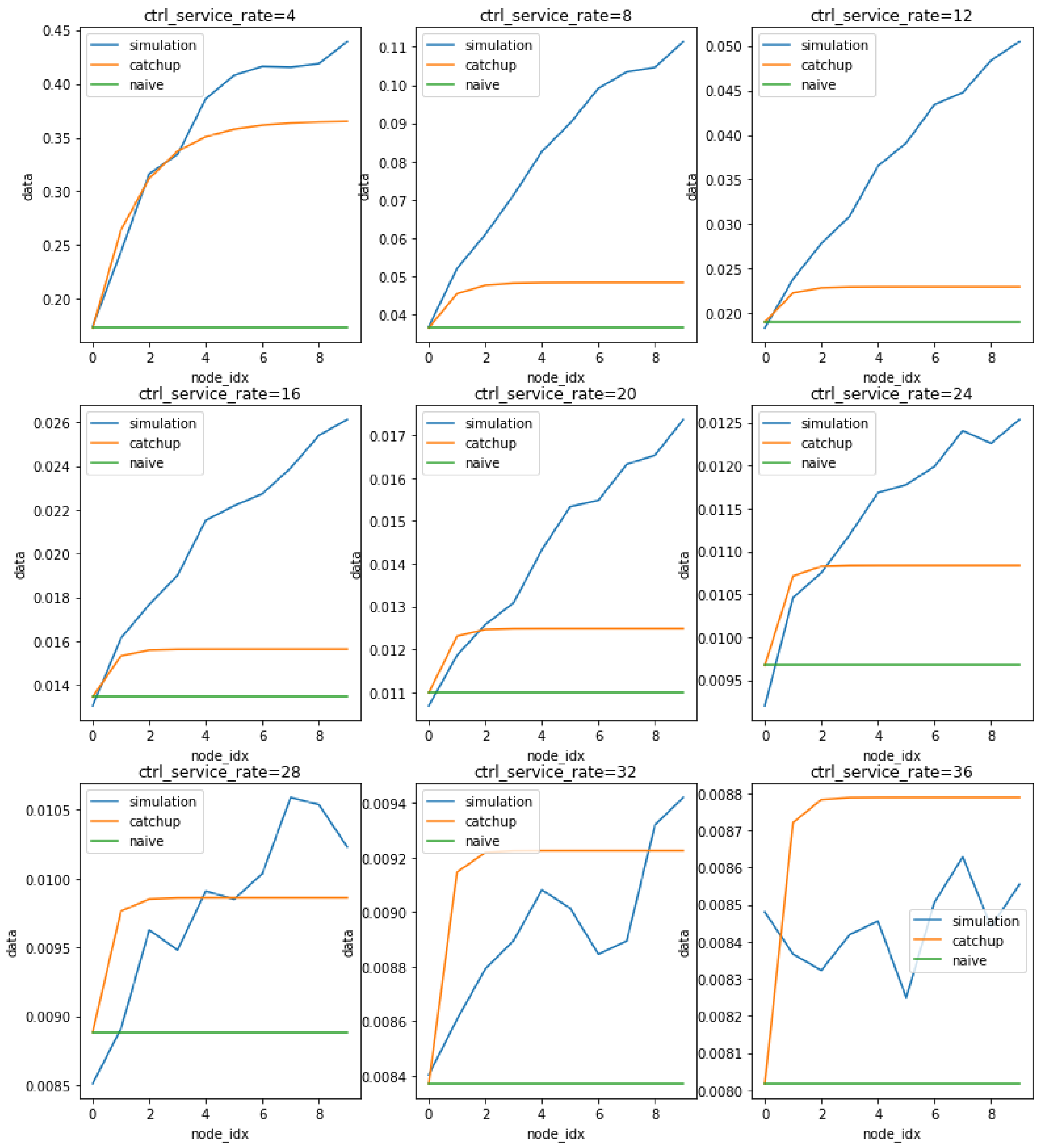

Figure 4.7: Comparing catchup and naive estimation of average waiting time of data packets at node with given index against simulation, with $\mu_{control\_node}$ from 4 to 36

## 4.6 Curve Fitting Approximation Using Partial Simulation

This section introduces a curve fitting approximation method, which uses polynomial functions to approximate the waiting time of data packets at intermediate nodes. On one hand, polynomial functions allow a faster convergence compared to catchup formulation. [1] On the other hand, it might be beneficial to enable performance estimation of long sequence networks using simulation of short sequences, considering that simulating networks with a long sequence of intermediate node may take a long time, or even be infeasible, while simulating networks with a short sequence of intermediate nodes is easier and more practical.

The procedure of approximate $N_{long}$ nodes using simulation of $N_{short}$ nodes is as follows:

1 Run simulations using $N_{short}$ intermediate nodes

2 Try to fit the curve of data packet waiting time with respect to node index (starting from 1 for the first node [2]) to the polynomial function: ($a, b, c$ are parameters)

$$\frac{a}{b+x} + c \tag{4.20}$$

3a If the fit is success, and the resulting function is monotonic increasing, input the sequence from 1 to $N_{long}$ to obtain the waiting time of intermediate nodes from the first node to the $N_{long}$-th node

3b If the fit failed, or the resulting function is not monotonic increasing, the network might already reached equilibrium, thus the waiting time of all the nodes is approximately the average waiting time of first $N_{short}$ nodes

[1]Other families of functions have been tried, but they don't perform as good as polynomial functions. Exponential functions converge too fast, while logarithmic functions converge too slow.

[2]The index starts from 1 instead of 0 to prevent divide by 0 issues when perform fitting

Table 4.3: Errors of different estimation methods with $\mu_{control\_node}$ from 4 to 40

Error is calculated as the sum of estimated waiting time at every node divided by the sum of simulated waiting time at every node.

| $\mu_{control\_node}$ | Curve Fit | Catchup | Naïve |
|---|---|---|---|
| 4 | -4.502% | 19.366% | 61.452% |
| 8 | -8.823% | 64.055% | 72.753% |
| 12 | -4.033% | 66.211% | 71.968% |
| 16 | -13.926% | 61.345% | 66.664% |
| 20 | 6.400% | 52.228% | 57.903% |
| 24 | 6.237% | 37.458% | 44.126% |
| 28 | 8.535% | 22.405% | 30.015% |
| 32 | -0.085% | 8.999% | 17.349% |
| 36 | 1.916% | 0.151% | 8.827% |
| 40 | -0.360% | -4.826% | 3.878% |
| Average | -0.864% | 32.739% | 43.494% |

The results of this simulation are shown in Fig. 4.8 and Fig. 4.9, using the first 20 nodes to estimate the waiting time at 100 nodes, using `curve_fit` from `scipy` library. [48] On average, the curve fit approximation has 7% over-estimation compared to the simulated results, based on the experiment of Fig. 4.9. A table summarizing the errors of different estimation methods with $\mu_{control\_node}$ can be found in Table 4.3. It can be seen that in most cases the curve fit approximation gives a reasonably well estimation.

This chapter analyzed the behavior of a Wireless Sensor Network using queueing theory, analytically solved the waiting time at the first node and proposed a curve fitting approximation to estimate the waiting time at subsequent nodes. However, even the curve fitting approximation performs better than the naive approximation using only the first node, it is far from perfect. For future research, more advanced approximation algorithms, such as deep learning may be applicable to this problem. Besides, some assumptions are too simple for real world usage, and it should be possible to model the performance of a system that is more practical.

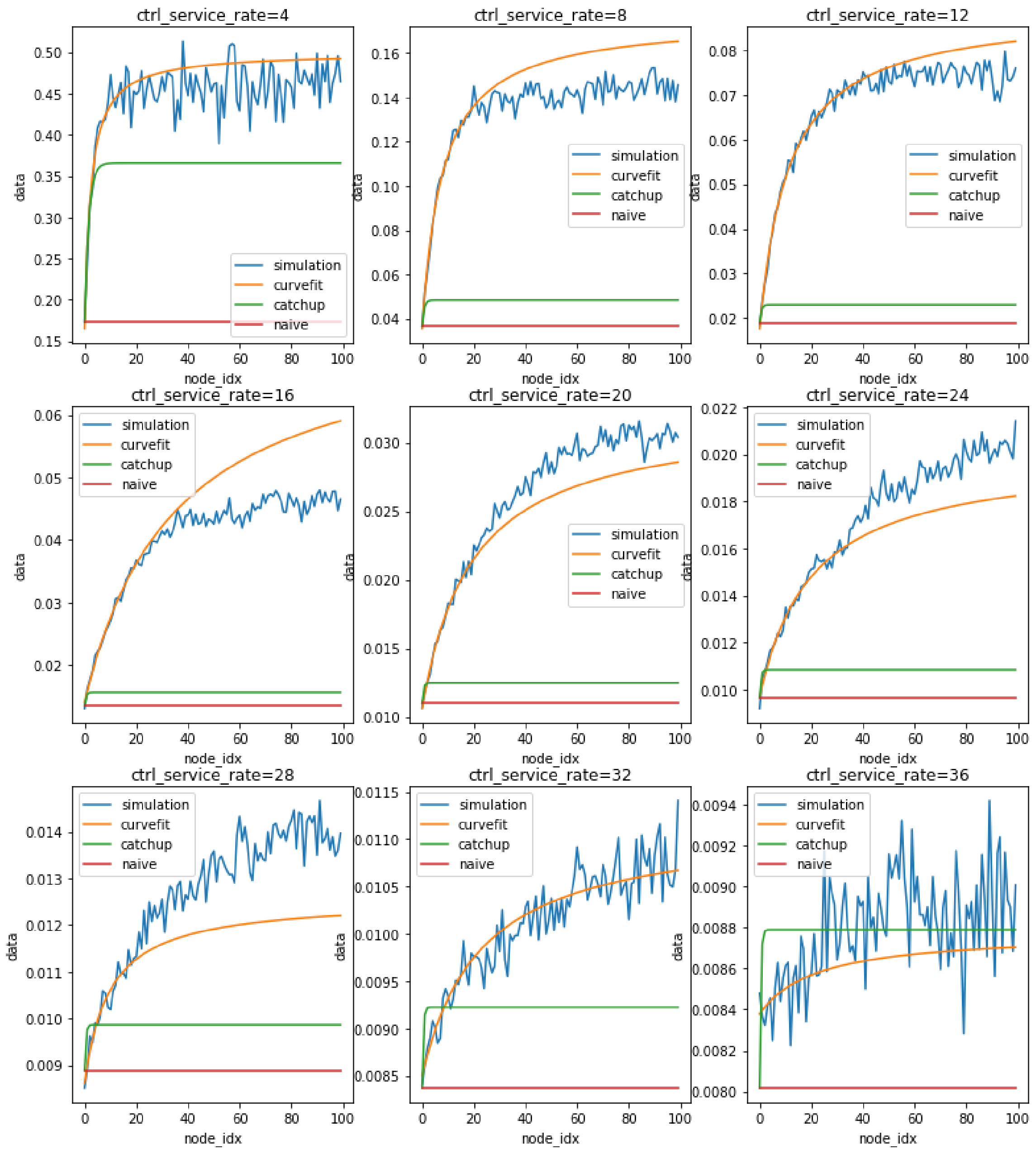

Figure 4.8: Comparing curve fit, catchup, naive estimation of average waiting time of data packets at node with given index against simulation, with $\mu_{control\_node}$ from 4 to 36

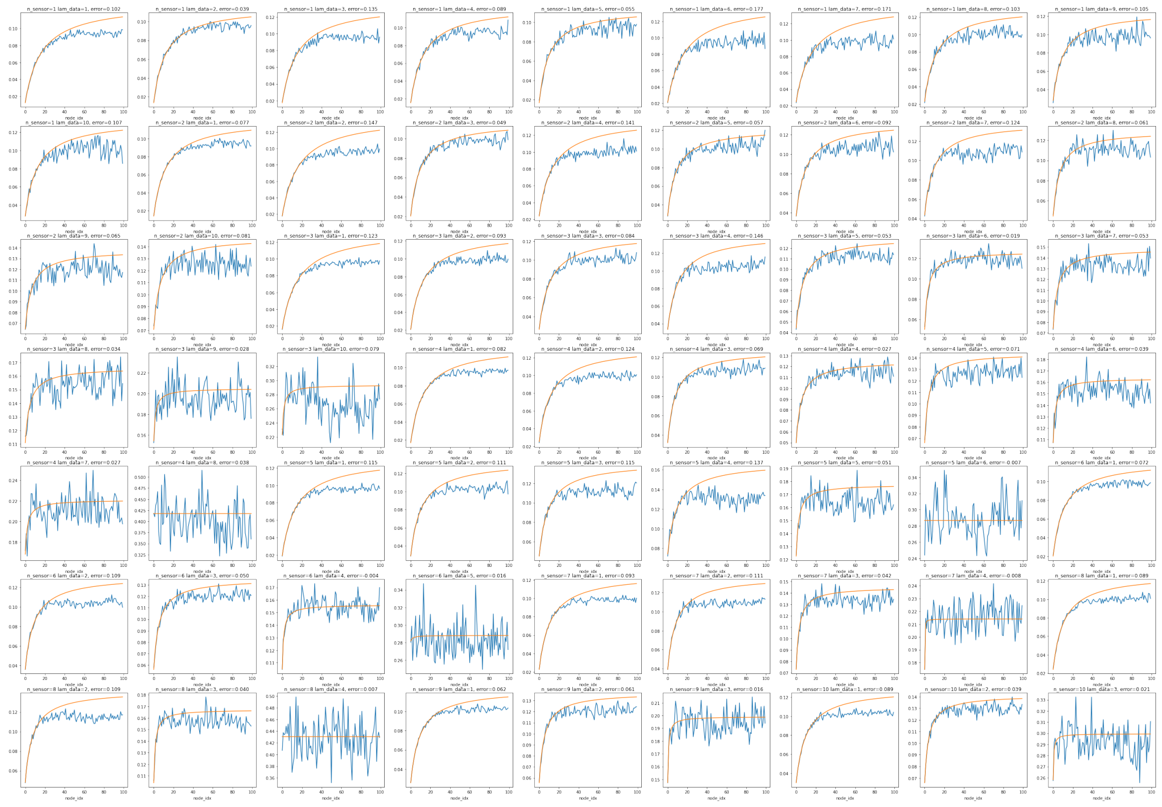

Figure 4.9: Comparing curve fit estimation against simulation, with $\mu_{data\_sensor}$ from 1 to 11 and $\lambda_{data}$ from 1 to 11.

Blue is the simulation and yellow is the curve fitting results. Error is calculated as the sum of estimated waiting time at every node divided by the sum of simulated waiting time at every node.

# Chapter 5

# Authenticate: Molecular Key, A Molecular-based Authentication System

After the discussion of data analysis and data acquisition, now we turn our attention to the security aspect. Multiple authentication methods already exist, and multi-factor authentication that combines them has shown great success in keeping information systems secured. However, when considering e-health applications, especially under the current COVID pandemic, it might be beneficial to incorporate health and environmental information into the security policy, extending security from virtual world to physical world. As an example, consider a hospital that denies COVID-contacted healthcare works accessing inpatient wards. Following this idea, this chapter proposes a novel molecular keys-based authentication system as an preliminary extension to existing ones, using molecules as data carriers while considering their types, concentration, and the arrival time by the receiver. A prototype is built to examine its validity, achieving a decoding accuracy of 86% for 3-bit sequences within a distance of 1m.

## 5.1 Introduction

Authentication is the process or action of verifying the identity of a user, process, or entity. To perform authentication, the user needs to supply evidence to support its identity claim, with every piece of the evidence called a factor. The most traditional yet commonly used authentication factor is the password, which is a string that only the user knows. Based on the source, authentication factors can be classified into three categories: knowledge (something the user knows, such as password, Personal Identification Number (PIN), and Short Message Service (SMS) code), ownership (something the user has, such as USB security token and Radio Frequency IDentification (RFID) smart card), and inherent characteristics (user's features, such as fingerprint, iris, and typing pattern). Besides using only one factor for authentication, multiple factors can also be used together to improve security. In multi-factor authentication, any missing or incorrect factor will cause the authentication process to fail; thus, the identity is only verified when all factors are correctly applied [32].

Although multiple authentication factors have been created to accommodate different use cases, they struggle to incorporate environmental and human health information. For example, in industrial biochemistry, current authentication methods are used to limit entering specific labs or areas to avoid contamination. However, individuals may be subjected to undesirable substances due to leakage or contamination, for example. Thus, we need a compatible authentication factor to detect such cases, which cannot be achieved using Electromagnetic (EM) based communication systems such as RFID, smart cards, and others.

While EM-based communication systems are having difficulties interacting with the physical world, MC uses molecules to carry information [36]; thus, it can work efficiently with several physical and biological environments [49]. Authentication via molecular keys allows utilizing environmental/health information to design a biocompatible security system and launch enhanced services. Indeed, motivated by the dynamic changes of authorization across time and places, emerging marketing applications, and evolving human activities, it is beneficial to consider another level of authentication across a molecular network. Molecular keys can be naturally chosen from a specific environmental molecular structure or a Volatile Organic Compound (VOC) associated with a human health and activity [50]. Moreover, the molecular keys can be artificially generated to support several everyday applications.

This section designs and builds a molecular key generator and detection system to be implemented as a side authentication factor. We build the prototype using a programmable VOC sprayer to generate molecular sequences that can be detected using a Photoionization Detector (PID). Then, we extensively evaluate multiple Machine Learning (ML) and Deep Learning (DL) classification models with different data normalization and feature engineering combinations to decode the transmitted molecular keys.

## 5.2 System Overview

This section gives an overview of the proposed prototype and associated system architecture used to generate and detect the molecular keys. As shown in Fig. 5.1, we show the system components, including the transmitter, receiver, modulation, as well as system testing to identify range limits.

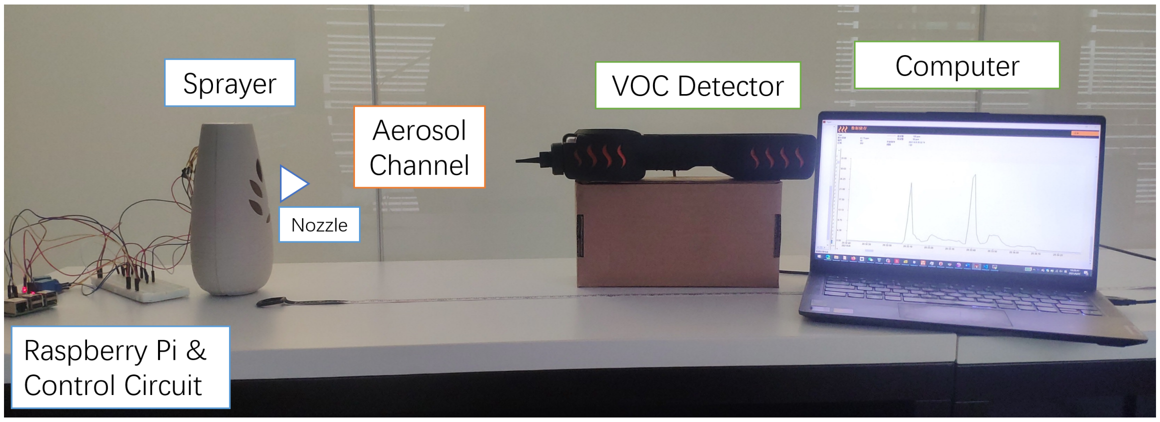

Figure 5.1: The molecular keys generation and detection prototype.

### 5.2.1 Transmitter and Receiver

We build the transmitter as a programmed VOC sprayer consisting of a mechanical sprayer connected to a small motor through gears. A molecular key is transmitted by the sprayer using a custom script on a connected Raspberry Pi that controls a motor via relays. Commodity air freshener cans can be used as VOC source, where Ethanol is the main component of the ingredients list. The motor rotates and presses the can for a certain amount of time to release some VOC molecules, then rotates reversely to stop releasing. The Raspberry Pi connectivity and flexibility is the key to manipulate the molecular keys generator via a remote connection. We use a photoionization VOC detector at the receiver side to measure the VOC concentration. This detector can detect a large spectrum of VOCs in a relatively short time (as fast as 1 sample per second) with a wide range of detection from 0.1 ppm to 5000 ppm. As a continuous stream of readings, the VOC concentration data can then be stored and retrieved from the detector for processing.

We built the whole system in a room with minimal human activity during the experiment, without any natural or artificial airflows. The distance and the alignment between the sprayer's sprinkler and the detector's nozzle can be adjusted.

### 5.2.2 Molecular Key Modulation

The molecular keys consist of a repeated sequence of ones and zeros modulated using the on-off keying scheme, where the sprayer is turned on only during the transmission of bit "1" duration. In the rest of the thesis, we design the system to generate 8 keys using 3 bits; however, an extended version while using more bits is also possible. To overcome the high system memory and slow propagation, as well as reducing the need for external synchronization between the transmitter and receiver, a simple communication protocol is used, as shown in Fig. 5.2. A pilot bit of "1" is prepended to the bit sequence to indicate the start of transmission. The bit duration $t_{\text{bit}}$ is assumed to be known at both ends. The transmitted symbol consists of the pilot bit and three key bits, which are periodically repeated after waiting idle period $t_{\text{idle}}$ to minimize the inter-symbol interference.

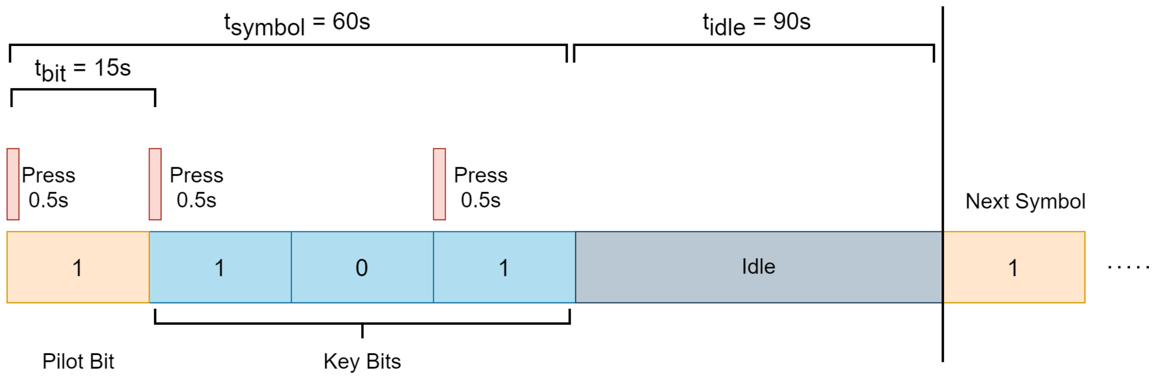

Figure 5.2: Modulation of the key 101

At the receiver side, the receiver operates under three states: idle, waiting, and receiving, as illustrated in Fig. 5.3. The receiving procedure starts at the idle state, monitoring the VOC concentration and looking for a drastic change to detect the pilot bit, which comes after some silence time that is greater than or equal to $t_{\text{idle}}$ by comparing the difference of current reading and the average reading of last 5 seconds to a predefined threshold. Then, it transits into a waiting state for a bit interval ($t_{\text{bit}}$) till the end of the pilot bit duration. After that, the receiver transits into a receiving state, in which it records the received VOC concentration for a symbol interval ($t_{\text{symbol}}$) to detect the transmitted key. After the transmission finishes, the receiver goes back to the idle state, and the VOC concentration segments are sent into a classification model to recover the transmitted bits.

### 5.2.3 System Testing

To assess the molecular key generation and detection system under the limited aerosol transmission channel, we conduct several experiments to define the system operating conditions and identify the limitations. The limitations mainly come from 2 aspects, signal strength and interference, as can be observed from the following experiments.

**Observed VOC Concentration.** We start by studying the VOC concentration observed at the receiver side after sending the key “101” as shown in Fig. 5.4. Despite the short VOC release press time, the received pulse width is wider than the press time

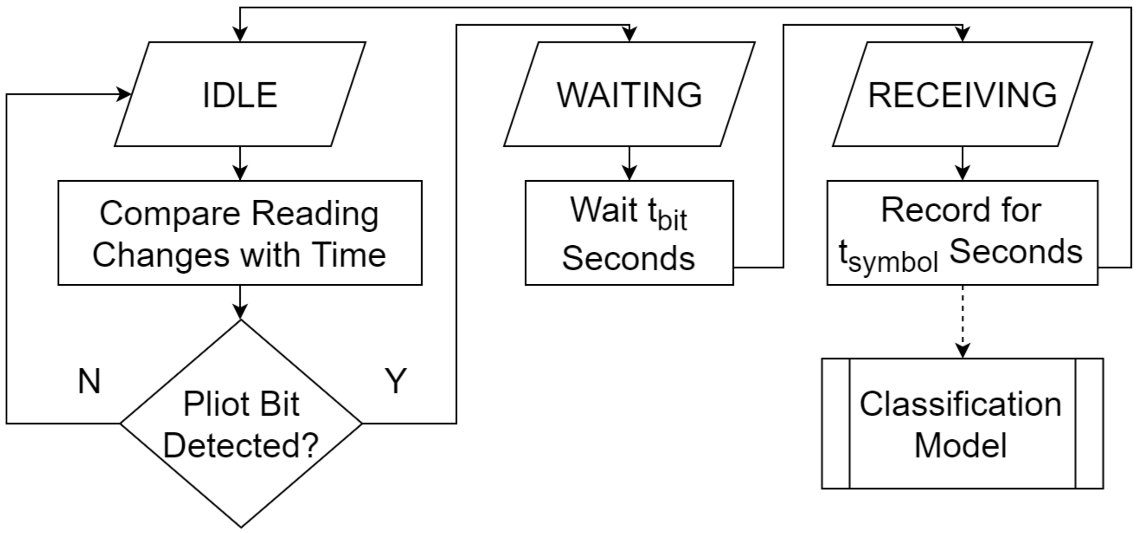

Figure 5.3: Receiver state transition.

due to the dispersion nature of the aerosol channel. The transmitted bits appear some spikes that can be identified and recognized as depicted in Fig. 5.4. Other fluctuations are due to environmental noises and channel turbulence. The previously mentioned reasons motivate us to reserve enough time between bit pressing time slots and also between the symbols, i.e., $t_{\text{idle}}$, to reduce the interference between adjacent bits and symbols.

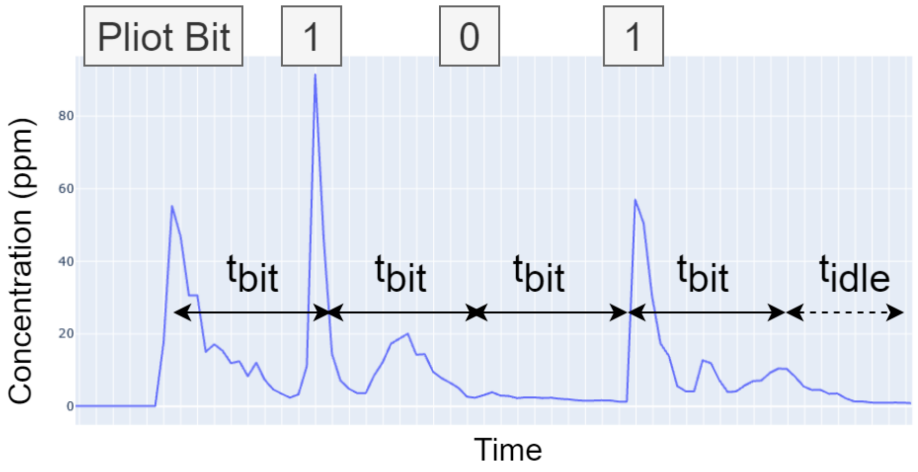

Figure 5.4: Received VOC concentration of key 101.

**Impact of Distance.** We send the key “111” with the sprayer and the detector aligned and measure the received signal strength in terms of the average VOC concentration versus different transmission range as shown in Fig. 5.5. We observe relatively high concentration measurements for small distances, allowing the keys to be detected efficiently. As the receiver moves further away, the measured VOC concentration decreases significantly due to the diffusion nature of the aerosol channel and the corresponding signal loss with the distance. According to the depicted results in Fig. 5.5 and the noise levels in Fig. 5.4, the adopted setup can operate satisfactorily for distances up to 1.5 m. However, at longer distances, the slow diffusion of the aerosol channel results in low values of received concentration compared to the environmental noise; thus, individual spikes may not be observed while the decoding is hardly possible.

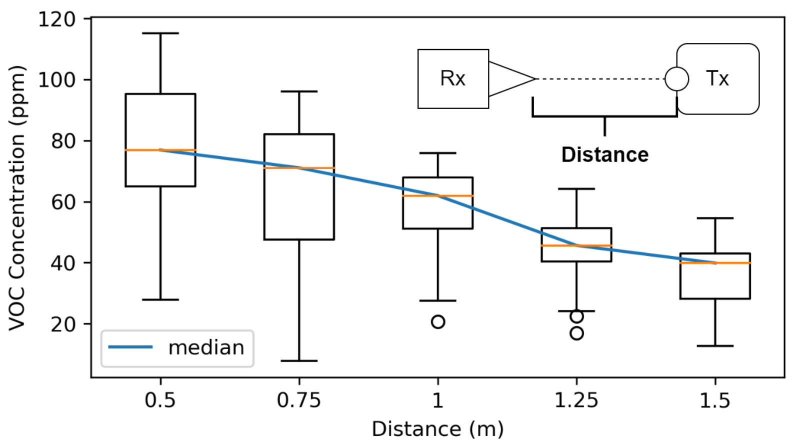

Figure 5.5: Effect of distance on VOC concentration.

**Impact of Alignment.** To understand the impact of alignment, we consider different angles between the spraying direction and the receiver at a distance of 0.7 m, using the same key "111". The value of angle quantifies the deviation of alignment. Since the released VOC mostly exists along the nozzle axis, the amount of VOC arrived at the detector decreases as the angle increases. Based on the results in Fig. 5.6 and the noise level shown in Fig. 5.4, the system can maintain a relatively good signal level within 20° angles range.

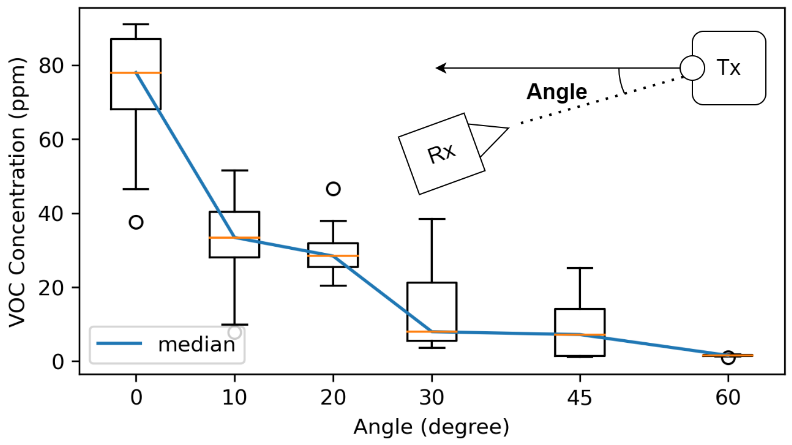

Figure 5.6: Effect of alignment on VOC concentration.

## 5.3 Problem Description

After generating different molecular keys and recording the VOC concentration at different time instants, the main problem becomes recovering these keys from the detected VOC concentration readings, similar to the depicted one in Fig. 5.4. The recovery process is done via two steps: First, segmenting the readings then, decoding the data.

**Segmentation.** After getting the VOC concentration readings for a time span of $T$ timesteps, $y_1, y_2, \ldots, y_T$, we isolate the reading segment, $\mathbf{Y}$, related to the key. The elements of $\mathbf{Y}$ are $Y_k[j]$, which represents the readings on $j$-th timestep of the $k$-th bit in the sequence.

**Decoding.** To decode $\mathbf{Y}$ data into the estimated key bits, we consider it a classification problem under variable channel and noise conditions. We perform the classification using learning algorithms that need extensive experimental training phases. Multiple classification models in ML can be applied to this problem, but some of them require larger dataset for better generalization. To address this problem efficiently, we first model the received bit concentration function based on several experimental measurements in the following section. Then, we use the model to generate a simulated dataset and train different classification learning models in Section 5.

## 5.4 VOC Concentration Modeling

In this section, we model the received VOC concentration by fitting theoretical models to our experimental data. We use the model proposed in [39] that has shown to be suitable for molecular channel characteristics.

**Modeling Single Release.** The received VOC concentration of a single transmitted pulse is modeled [39] as a function in the time $t$ as,

$$\lambda(t) = \begin{cases} \kappa\sqrt{\frac{c}{2\pi t^3}}\exp\left[-\frac{c(t-\mu)^2}{2\mu^2 t}\right] & t > 0 \\ 0 & t \leq 0 \end{cases} \tag{5.1}$$

where $\kappa$ is a proportionality constant, $c$ and $\mu$ are channel parameter, which can be found by fitting the curve to experimental data.

**Fitting Single Release.** To find $\kappa$, $c$ and $\mu$, we conducted 30 single-release experiments for aligned sprayer and detector while allow an idle time of 120 seconds between consecutive presses to avoid signal interference. Then, we use a simulated annealing algorithm (implemented in [51]) to fit the parameters, by minimizing Root-mean-square Error (RMSE) between values predicted by the model and the experiment, over the time length 45 seconds at a sampling rate of 1 sample/sec. The parameters are found to be: $\kappa = 408$, $c = 71.2$ and $\mu = 11.9$. Fig. 5.7 shows the 30 received signal of single press versus time overlayed on the fitted model, which verifies the good approximation.

**Modeling Sequence Release.** Following the fitted model for the single release, we model the stream of molecular bits and develop an appropriate model. The summation of delayed versions of $\lambda(t)$ follows a Poisson distribution [39]; thus, the distribution of the received VOC concentration is expressed as,

$$Y_k[j] \sim \mathscr{P}\left(\sum_{i=0}^{k} x_{k-i}\lambda_i[j] + \eta\right) \tag{5.2}$$

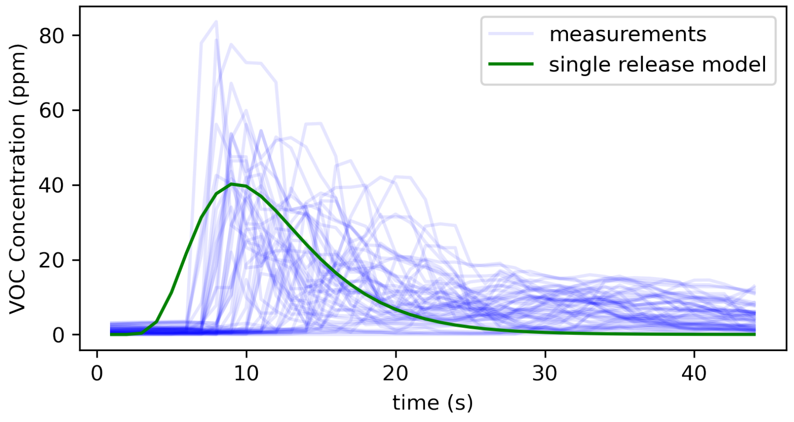

Figure 5.7: Comparison between several measurements of the received single release pulse and the adopted model.

where $\eta$ represents an independent additive Poisson noise coming from background or the receiver, $\mathscr{P}(\xi) = \frac{\xi^y e^{-\xi}}{y!}$ is the Poisson distribution function with a parameter $\xi$ and $y$ is the measured value. By interpreting the parameter $\xi$ as the amount of VOC molecules in the air near the detector, sampling from the Poisson distribution simulates the measuring of VOC concentration since Poisson distribution describes the probability of random events happening in a given time, and molecule being captured by the detector is such an event. $\lambda_i[j]$ is the response at the $j$-th timestep of the current bit due to the $i$-th bit in the transmission and is expressed as

$$\lambda_i[j] = \lambda(\frac{i \cdot \omega \cdot \mathrm{t_{bit}} + j}{\omega}) \tag{5.3}$$

with $\omega$ denoting the sampling rate.

## 5.5 Retrieving Molecular Keys

Throughout this section, we detail the implementation of the keys decoding process from the received VOC concentrations with the help of the previously discussed channel characterization simulation model. Covering all possible fluctuations by experiments would be impossible; thus, the simulated dataset can help the decoding models access plenty of generalized results. By viewing the decoding problem as a time series classification problem, various model configurations can be built using different models, normalization, and feature engineering techniques.

**Building Simulation Dataset.** We developed a customized script to simulate the transmission of different keys sequences considering randomization of time alignment, concentration amplitude, and environmental noises to simulate the VOC diffusion more realistically. To this end, we created a simulated dataset of size 10000 and adopted the segmentation procedure to capture the key sequence information.

**Normalization and Feature Engineering.** Before processing the received data, it is necessary to normalize them to improve the predictability and model robustness. We use two normalization methods: the Z-normalization, which normalizes the data to have zero-mean and unity standard deviation values, and the min-max normalization, which normalizes the data between 0 and 1. Regarding feature engineering, we use two techniques: slope, which calculates the difference between each time step, and summary, which computes different statistics such as the maximum, minimum, variance, median, and mean.

**Methods.** As a univariate time series classification problem, several ML classification methods are applicable. All these methods can be categorized considering two perspectives, as shown in Table 5.1. Firstly, the methods can be classified as to whether DL can be involved or just classic ML. Secondly, they can be classified according to whether the time semantic is preserved or not, i.e., whether the input is seen as a pure vector or a time series.

Here are some additional explanations to Table 5.1:

- CNN is classified as "Keep Time Semantic" as it performs convolution along the time axis, thus retaining the sequence nature of a time series.

[1]Time Series KNN with Dynamic Time Warping (DTW), implemented in [58].
[2]Time Series Forest, [59] implemented in [58].
[3]Random Interval Spectral Ensemble, [60] implemented in [58].
[4]Fully Convolutional Neural Network and a relatively deep Residual Network [61].

Table 5.1: Methods Summary

| | Preserve Time Semantic | Discard Time Semantic |
|---|---|---|
| Classic ML | TS-KNN[1]<br>TS-Forest[2]<br>RISE[3] | KNN[52]*<br>SVM[53]*<br>Naive Bayes*<br>Decision Tree*<br>Random Forest* |
| Deep Learning | CNN (FCN/ResNet [4])*<br>RNN (LSTM[54]/GRU[55] )*<br>BiRNN[56] (LSTM/GRU)* | MLP[57] * |

- For Bidirectional RNNs, a variant using 3 Bidirectional layers (denoted by "3Bi") is also tried.

- For MLP and CNN Methods, the implementation is based on a review paper by H. Fawaz etal [62]. CNN methods are chosen as the top 3 ranked methods according to their pairwise ranking on univariate time series classification.

For the methods with an asterisk, besides directly taking the time series as input, a variant which takes both the input sequence and its summary (as described in Feature Engineering) is also tried. For Classic ML methods, the summary is simply concatenated along with the input sequence. For DL methods, the summary is inputted as another branch, which first passes 2 fully connected layers, then a dropout layer, finally concatenated with the output from the sequence branch, before the final fully connected layer and the output layer, as shown in Fig. 5.8.

**Training.** To make a fair comparison, most models are trained in the same fashion:

- Classic ML methods: First run a grid search with 5-fold cross validation on the simulated dataset to find the best parameters, then the model is trained using the whole simulated dataset and tested on the real dataset.

- DL methods: All models use an Adam optimizer with a learning rate of 1e-3. Models can be trained at most 200 epochs on the simulated dataset, but an

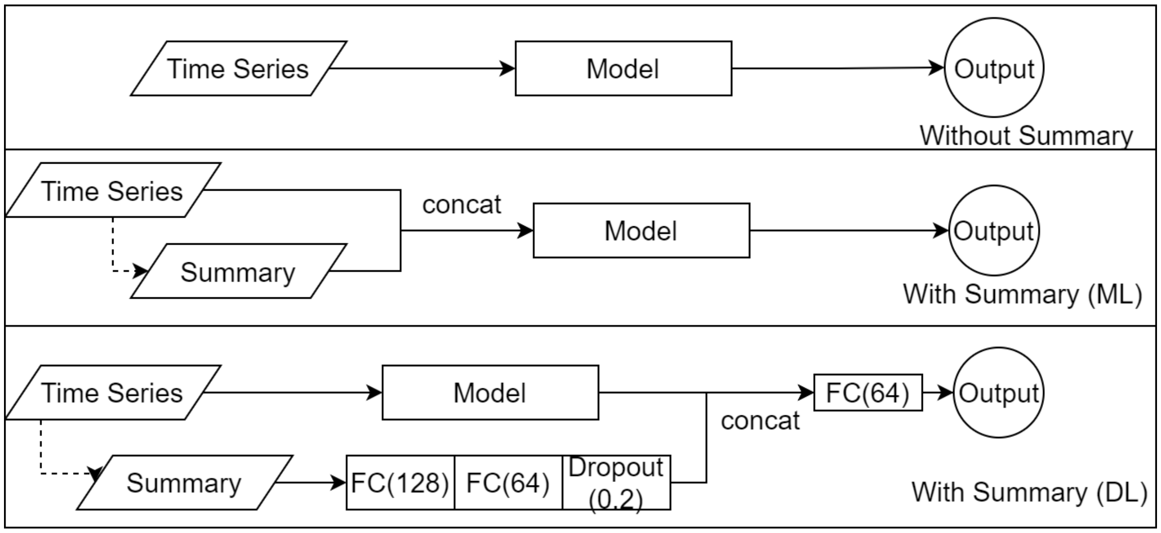

Figure 5.8: Architecture of classification models with and without summary.

early stop callback is registered on the validation loss (validation set taken from simulated dataset), so the training will stop if the loss does not decrease in 10 epochs.

Some specific measures are used when the general training methods do not work well, as stated below.

- For Time Series ML methods, running grid search with 5-fold CV takes a long time, so only 10% of the simulated dataset for this stage. Later the model is still trained using the whole simulated dataset and the best parameters found.
- For FCN in CNN, AdaDelta optimizer is used instead of Adam [62].

## 5.6 System Performance Evaluation

In this section, we evaluate the decoding performance of different classification learning methods using experimentally measured data to find the best methods. We use two metrics known as accuracy and class-weighted F1, which are suitable for our multi-class classification problem. The accuracy is defined as the percentage of correct predictions on all predictions. While the class-weighted F1 is defined as the sum of F1 scores in each class weighted by the class size, which is expressed as,

$$\text{Class Weighted F1} = \sum_{i=0}^{C} \frac{N_i}{N} \text{F1}_i, \tag{5.4}$$

where $C$ denotes the number of classes, $N$ denotes the records number, $N_i$ denotes the records number in the class $i$, and $F1_i$ denotes the F1 value of the class $i$, which is found from

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.5}$$

where the precision and recall are computed using True Positive (TP), False Positive (FP) and False Negative (FN) as,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{5.6}$$

Table 5.2: Top 5 Methods

| Rank | ML/DL | Method | Normalization | Feature Engineering | | Simulated Dataset | | Sequence Release Dataset | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Slope | Summary | Accuracy | Weighted F1 | Accuracy | Weighted F1 |
| 1 | DL | BiLSTM | none | ✓ | | 0.974 | 0.975 | 0.864 | 0.868 |
| 2 | DL | 3BiGRU | znorm | ✓ | ✓ | 0.978 | 0.978 | 0.818 | 0.818 |
| 3 | ML | NaiveBayes | none | ✓ | | 0.658 | 0.661 | 0.818 | 0.816 |
| 4 | ML | kNN | none | | | 0.980 | 0.980 | 0.818 | 0.814 |
| 5 | DL | GRU | minmax | ✓ | ✓ | 0.973 | 0.973 | 0.795 | 0.790 |

We use the sprayer to transmit all possible 3-bit digital sequences, *i.e.*, from 000 to 111, following the modulation and protocol defined previously. After receiving the data at the detector side, the corresponding segments are isolated for each bit sequence, and then different ML classification methods are tested on 44 segments. Fig. 5.9 shows that the simulated data (in blue) is able to capture the characteristics of real data (in red) well. To quantify the performance of different ML classification methods, we computed the accuracy and weighted F1 metric for all methods using different normalization and feature engineering. Table 5.2 lists the best 4 methods while showing the adopted normalization and feature engineering. Among all methods, the Single Layer Bidirectional LSTM achieves the highest accuracy of 0.864.

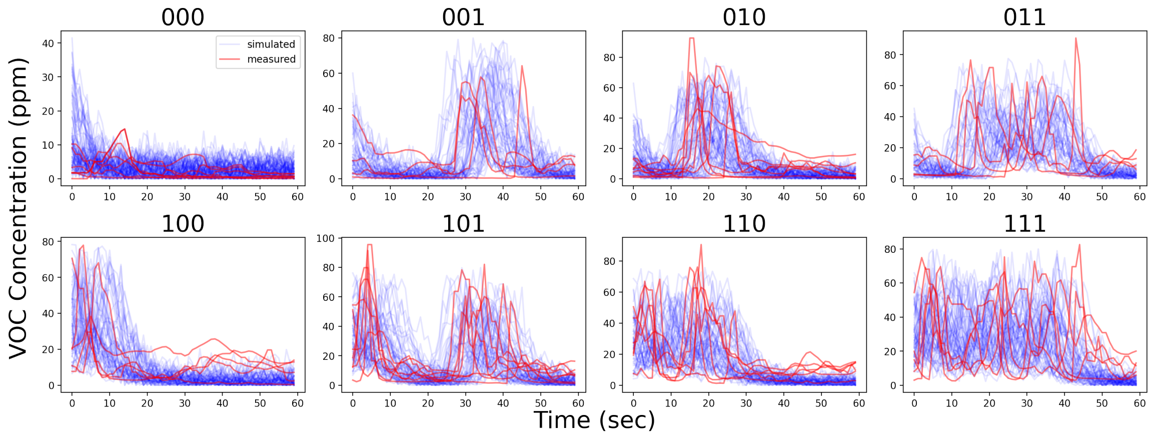

Figure 5.9: Comparison between the simulated datasets and the measurements of received key sequences.

## 5.7 Discussion

This section discusses possible applications motivated by the unique features and advantages of the proposed molecular key-based systems. Then, we highlight some challenges that should be addressed in future system implementations.

### 5.7.1 Potential Use Cases

**Biochemical Applications.** Although it is possible to implement the existing authentication methods to grant or deny access based on visited locations, it is not possible to use the contact of a specific substance to trigger these radio-frequency-based systems. However, by integrating appropriate MC detectors, such as PID, into handheld devices, the system can monitor the nearby biochemical compounds and use the predefined molecular keys (type and concentration) to check the authentication and grant permission.

In several industrial locations, it is either unsafe to use radio-frequency-based systems to avoid hazards (e.g., fire) or inadequate to use them in some environments such as tubes. Therefore, generating artificial compatible molecular keys with appropriate fluid flows can safely authenticate industrial processes and deal with different chemical substances.

**Human-centric Applications.** Human health and dietary habits can be easily monitored by the exhaled VOC [50]; thus, we can use the proposed molecular-keys-based systems for authentication and monitoring daily human activities. For example, the system can limit some human activities if the alcohol level in the breath exceeds a specific threshold. Moreover, the exhaled VOC profile can be studied, trained, and used to identify viral infection particles and possibly use such information to develop pandemic mitigation authentication policies. Also, the system can allow practicing sports or using gym equipment if the health condition is suitable based on the exhaled VOC measurements.

**Marketing Applications.** It is possible to use molecular keys for marketing purposes to register rewards, which can be used to get benefits such as accessing high-speed networks and special VIP zones. Specifically, since all marketing malls use air fresheners, we propose using artificial molecular keys impeded in them. Then, the customers can collect the keys by visiting the stores or specific supermarket lanes, which requires spending more time thanks to the slow propagation features of MC channels. Thus, the merchants can have more chances to show their goods and attract customers.

### 5.7.2 Implementation Challenges

To implement the molecular-keys-based systems in our everyday life applications, we should build a robust system that can operate under different conditions and mitigate several challenges. In the following, we summarize the main issues that can impede the implementation of the proposed system.

**Dynamic Environments.** The system can be affected by ambient environmental circumstances such as ventilation, interfered VOCs, and pedestrian flows, which should be considered in system modeling. Also, appropriate system implementation and detection algorithms need to be investigated to mitigate these side effects.

**Regulatory Compliance.** The systems that use artificial molecular keys should consider carefully any conditions regarding the environment. The adopted VOCs in the industrial environment should meet safety constraints and do not affect the production quality. Regarding the usage in public areas, the circulated air should be suitable for humans without causing any health problems.

**Scalability.** Implementing the system on a large-scale requires considering the design of long molecular keys, which needs using appropriate system components, protocol design, and detection algorithms. Also, accommodating multiple receiver and enabling the broadcasting feature are important feature that requires careful consideration such as different range and directions.

This chapter considered an authentication factor utilizing MC, named Molecular Key. Compared to traditional authentication factors, using molecules as carrier enables authentication process to interact with physical world easier, creating new possibilities and providing more flexibility to designing security systems. A prototype was built to show the feasibility of purposed system, using a sprayer the transmitter and a VOC detector as the receiver. Different ML and DL techniques were considered for recovering the bits transmitted from the VOC concentration reading stream. Advantages and current limitations are discussed, with potential applications in multiple domains.

# Chapter 6

# Conclusion

In this thesis, we thoroughly investigated the 3 major parts in building e-health applications: analyze, acquire, and authenticate, covering the recommended guidelines along with concrete examples. We showed that applying deep learning to sleep stage prediction problems can be beneficial, and introducing federated learning for privacy preservation did not significantly impact the model's performance. We observed that it is possible to estimate the delay in sensor networks despite its internal complexity and randomness. We found that using VOC-based molecular keys for authentication is not only possible, but may provide additional security guarantees in certain use cases. We believe that these work can help build a more concrete foundation for e-health applications.

With that said, many limitations still need to be addressed in future works. The latency model needs better mathematical functions beyond simple curve fitting. The time series labelling model used for sleep stage prediction tasks may be further refined and applied to other similarly structured problems. The amount of information that can be carried by the Molecular Keys are severely limited. We hope future research will solve these problems, as well as open new possibilities for the framework we proposed.

# REFERENCES

[1] J. Lockman, R. S. Fisher, and D. M. Olson, "Detection of seizure-like movements using a wrist accelerometer," *Epilepsy & Behavior*, vol. 20, no. 4, pp. 638–641, 2011.

[2] D. J. Wile, R. Ranawaya, and Z. H. Kiss, "Smart watch accelerometry for analysis and diagnosis of tremor," *Journal of neuroscience methods*, vol. 230, pp. 1–4, 2014.

[3] R. Powers, M. Etezadi-Amoli, E. M. Arnold, S. Kianian, I. Mance, M. Gibiansky, D. Trietsch, A. S. Alvarado, J. D. Kretlow, T. M. Herrington *et al.*, "Smartwatch inertial sensors continuously monitor real-world motor fluctuations in parkinson's disease," *Science translational medicine*, vol. 13, no. 579, 2021.

[4] O. Walch, Y. Huang, D. Forger, and C. Goldstein, "Sleep stage prediction with raw acceleration and photoplethysmography heart rate data derived from a consumer wearable device," *Sleep*, vol. 42, no. 12, p. zsz180, Dec. 2019. [Online]. Available: https://doi.org/10.1093/sleep/zsz180

[5] A. Bhandary, G. A. Prabhu, V. Rajinikanth, K. P. Thanaraj, S. C. Satapathy, D. E. Robbins, C. Shasky, Y.-D. Zhang, J. M. R. Tavares, and N. S. M. Raja, "Deep-learning framework to detect lung abnormality–a study with chest x-ray and lung ct scan images," *Pattern Recognition Letters*, vol. 129, pp. 271–278, 2020.

[6] J. Xu, "Distance-based protein folding powered by deep learning," *Proceedings of the National Academy of Sciences*, vol. 116, no. 34, pp. 16 856–16 865, 2019.

[7] G. Eysenbach, "What is e-health?" *Journal of Medical Internet Research*, vol. 3, no. 2, p. e833, Jun. 2001, company: Journal of Medical Internet Research Distributor: Journal of Medical Internet Research Institution: Journal of Medical Internet Research Label: Journal of Medical Internet Research Publisher: JMIR Publications Inc., Toronto, Canada. [Online]. Available: https://www.jmir.org/2001/2/e20

[8] WHO Global Observatory for eHealth, "mHealth: new horizons for health through mobile technologies: second global survey on eHealth,"

World Health Organization, Tech. Rep., 2011, iSBN: 9789244564257 ISSN: 2305-0934 (Online) number-of-pages: viii, 102. [Online]. Available: https://apps.who.int/iris/handle/10665/44607

[9] S. Berrouiguet, M. M. Perez-Rodriguez, M. Larsen, E. Baca-García, P. Courtet, and M. Oquendo, "From eHealth to iHealth: Transition to Participatory and Personalized Medicine in Mental Health," *Journal of Medical Internet Research*, vol. 20, no. 1, p. e7412, Jan. 2018, company: Journal of Medical Internet Research Distributor: Journal of Medical Internet Research Institution: Journal of Medical Internet Research Label: Journal of Medical Internet Research Publisher: JMIR Publications Inc., Toronto, Canada. [Online]. Available: https://www.jmir.org/2018/1/e2

[10] J. Budd, B. S. Miller, E. M. Manning, V. Lampos, M. Zhuang, M. Edelstein, G. Rees, V. C. Emery, M. M. Stevens, N. Keegan, M. J. Short, D. Pillay, E. Manley, I. J. Cox, D. Heymann, A. M. Johnson, and R. A. McKendry, "Digital technologies in the public-health response to COVID-19," *Nature Medicine*, vol. 26, no. 8, pp. 1183–1192, Aug. 2020, bandiera_abtest: a Cg_type: Nature Research Journals Number: 8 Primary_atype: Reviews Publisher: Nature Publishing Group Subject_term: Health care;Viral infection Subject_term_id: health-care;viral-infection. [Online]. Available: https://www.nature.com/articles/s41591-020-1011-4

[11] J. Corkery, "Google and Harris Poll Healthcare interoperability survey." [Online]. Available: https://cloud.google.com/blog/topics/healthcare-life-sciences/google-and-harris-poll-healthcare-interoperability-survey/

[12] M. L. Taylor, E. E. Thomas, C. L. Snoswell, A. C. Smith, and L. J. Caffery, "Does remote patient monitoring reduce acute care use? A systematic review," *BMJ Open*, vol. 11, no. 3, p. e040232, Mar. 2021, publisher: British Medical Journal Publishing Group Section: Health services research. [Online]. Available: https://bmjopen.bmj.com/content/11/3/e040232

[13] L. P. Malasinghe, N. Ramzan, and K. Dahal, "Remote patient monitoring: a comprehensive study," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 1, pp. 57–76, Jan. 2019. [Online]. Available: http://link.springer.com/10.1007/s12652-017-0598-x

[14] K. Wac, R. Bults, B. van Beijnum, I. Widya, V. Jones, D. Konstantas, M. Vollenbroek-Hutten, and H. Hermens, "Mobile patient monitoring: The MobiHealth system," in *2009 Annual International Conference of the IEEE Engi-*

*neering in Medicine and Biology Society*, Sep. 2009, pp. 1238–1241, iSSN: 1558-4615.

[15] L. Skorin-Kapov and M. Matijasevic, "Analysis of QoS Requirements for e-Health Services and Mapping to Evolved Packet System QoS Classes," *International Journal of Telemedicine and Applications*, vol. 2010, pp. 1–18, 2010. [Online]. Available: http://www.hindawi.com/journals/ijta/2010/628086/

[16] L. Qiao and P. Koutsakis, "Guaranteed Bandwidth Allocation and QoS support for Mobile Telemedicine Traffic," in *2008 IEEE Sarnoff Symposium*, Apr. 2008, pp. 1–5.

[17] C. Li, X. Hu, and L. Zhang, "The IoT-based heart disease monitoring system for pervasive healthcare service," *Procedia Computer Science*, vol. 112, pp. 2328–2334, Jan. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050917316745

[18] J. Cullen, W. Gaasch, D. Gagliano, J. Goins, and R. Gunawardane, "Wireless mobile telemedicine: En-route transmission with dynamic quality-of-service management," in *National Library of Medicine Symposium on Telemedicine and Telecommunications: Options for the New Century*, 2001.

[19] K. Wac and R. Bults, "Performance evaluation of a transport system supporting the mobihealth banip: Methodology and assessment," Ph.D. dissertation, 2004, iD: unige:73317. [Online]. Available: https://archive-ouverte.unige.ch/unige:73317

[20] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "Joint Classification and Prediction CNN Framework for Automatic Sleep Stage Classification," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 5, pp. 1285–1296, May 2019, conference Name: IEEE Transactions on Biomedical Engineering.

[21] S. Kanwal, M. Uzair, H. Ullah, S. D. Khan, M. Ullah, and F. A. Cheikh, "An Image Based Prediction Model for Sleep Stage Identification," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 1366–1370, iSSN: 2381-8549.

[22] Y. Zhang, Z. Yang, K. Lan, X. Liu, Z. Zhang, P. Li, D. Cao, J. Zheng, and J. Pan, "Sleep Stage Classification Using Bidirectional LSTM in Wearable Multi-sensor Systems," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 443–448.

[23] A. Koushik, J. Amores, and P. Maes, "Real-Time Sleep Staging using Deep Learning on a Smartphone for a Wearable EEG," *arXiv:1811.10111 [cs, eess, q-bio]*, Nov. 2018, arXiv: 1811.10111. [Online]. Available: http://arxiv.org/abs/1811.10111

[24] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, Nov. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835220305532

[25] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated Learning for Mobile Keyboard Prediction," *arXiv:1811.03604 [cs]*, Feb. 2019, arXiv: 1811.03604. [Online]. Available: http://arxiv.org/abs/1811.03604

[26] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. Van Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards Federated Learning at Scale: System Design," *arXiv:1902.01046 [cs, stat]*, Mar. 2019, arXiv: 1902.01046. [Online]. Available: http://arxiv.org/abs/1902.01046

[27] N. Kourtellis, K. Katevas, and D. Perino, "FLaaS: Federated Learning as a Service," *Proceedings of the 1st Workshop on Distributed Machine Learning*, pp. 7–13, Dec. 2020, arXiv: 2011.09359. [Online]. Available: http://arxiv.org/abs/2011.09359

[28] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three Approaches for Personalization with Applications to Federated Learning," *arXiv:2002.10619 [cs, stat]*, Jul. 2020, arXiv: 2002.10619. [Online]. Available: http://arxiv.org/abs/2002.10619

[29] "TensorFlow Federated." [Online]. Available: https://www.tensorflow.org/federated?hl=zh-cn

[30] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated Learning for Healthcare Informatics," *Journal of Healthcare Informatics Research*, vol. 5, no. 1, pp. 1–19, Mar. 2021. [Online]. Available: https://doi.org/10.1007/s41666-020-00082-4

[31] S. R. Pfohl, A. M. Dai, and K. Heller, "Federated and Differentially Private Learning for Electronic Health Records," *arXiv:1911.05861 [cs, stat]*, Nov. 2019, arXiv: 1911.05861. [Online]. Available: http://arxiv.org/abs/1911.05861

[32] M. H. Barkadehi, M. Nilashi, O. Ibrahim, A. Zakeri Fardi, and S. Samad, "Authentication systems: A literature review and classification," *Telematics and Informatics*, vol. 35, no. 5, pp. 1491–1511, Aug. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736585318301400

[33] L. Lu, J. Yu, Y. Chen, H. Liu, Y. Zhu, Y. Liu, and M. Li, "Lippass: Lip reading-based user authentication on smartphones leveraging acoustic signals," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 1466–1474.

[34] J. Thorpe, P. C. Van Oorschot, and A. Somayaji, "Pass-thoughts: authenticating with our minds," in *Proceedings of the 2005 workshop on New security paradigms*, 2005, pp. 45–56.

[35] I. Švogor and T. Kišasondi, "Two factor authentication using eeg augmented passwords," in *Proceedings of the ITI 2012 34th International Conference on Information Technology Interfaces*, 2012, pp. 373–378.

[36] T. Nakano, T. Suda, M. Moore, R. Egashira, A. Enomoto, and K. Arima, "Molecular communication for nanomachines using intercellular calcium signaling," in *5th IEEE Conference on Nanotechnology, 2005.*, Jul. 2005, pp. 478–481 vol. 2, iSSN: 1944-9399.

[37] N. Farsad, W. Guo, and A. W. Eckford, "Tabletop Molecular Communication: Text Messages through Chemical Signals," *PLoS ONE*, vol. 8, no. 12, p. e82935, Dec. 2013. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0082935

[38] N. Farsad and A. Goldsmith, "Detection Algorithms for Communication Systems Using Deep Learning," *arXiv:1705.08044 [cs]*, Jul. 2017, arXiv: 1705.08044. [Online]. Available: http://arxiv.org/abs/1705.08044

[39] ——, "Neural Network Detection of Data Sequences in Communication Systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5663–5678, Nov. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8454325/

[40] "CDC - Data and Statistics - Sleep and Sleep Disorders," Sep. 2021. [Online]. Available: https://www.cdc.gov/sleep/data_statistics.html

[41] "2020 Sleepiness and Low Levels of Action," Mar. 2020. [Online]. Available: https://www.sleepfoundation.org/professionals/sleep-america-polls/2020-sleepiness-and-low-action

[42] D. E. Bild, D. A. Bluemke, G. L. Burke, R. Detrano, A. V. Diez Roux, A. R. Folsom, P. Greenland, D. R. JacobsJr., R. Kronmal, K. Liu, J. C. Nelson, D. O'Leary, M. F. Saad, S. Shea, M. Szklo, and R. P. Tracy, "Multi-Ethnic Study of Atherosclerosis: Objectives and Design," *American Journal of Epidemiology*, vol. 156, no. 9, pp. 871–881, Nov. 2002. [Online]. Available: https://doi.org/10.1093/aje/kwf113

[43] S. Mahmud, "UNet-Model-Builder-Tensorflow-Keras," Nov. 2021, original-date: 2021-08-10T10:01:55Z. [Online]. Available: https://github.com/Sakib1263/UNet-Segmentation-AutoEncoder-1D-2D-Tensorflow-Keras

[44] "Federated Learning for Image Classification | TensorFlow Federated." [Online]. Available: https://www.tensorflow.org/federated/tutorials/federated_learning_for_image_classification?hl=zh-cn

[45] Y. Chen, T. Farley, and N. Ye, "QoS Requirements of Network Applications on the Internet," p. 22.

[46] S. Butner and M. Ghodoussi, "Transforming a surgical robot for human telesurgery," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 818–824, Oct. 2003. [Online]. Available: http://ieeexplore.ieee.org/document/1236755/

[47] M. Harchol-Balter, *Performance modeling and design of computer systems: queueing theory in action.* Cambridge University Press, 2013.

[48] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[49] D. T. McGuiness, S. Giannoukos, S. Taylor, and A. Marshall, "Analysis of Multi-Chemical Transmission in the Macro-Scale," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 6, no. 2, pp. 93–106, Nov. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9151989/

[50] M. Khalid, O. Amin, S. Ahmed, B. Shihada, and M.-S. Alouini, "Communication through breath: Aerosol transmission," *IEEE Communications Magazine*, vol. 57, no. 2, pp. 33–39, 2019.

[51] "microsoft/nni," Jun. 2021, original-date: 2018-06-01T05:51:44Z. [Online]. Available: https://github.com/microsoft/nni

[52] N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992, publisher: [American Statistical Association, Taylor & Francis, Ltd.]. [Online]. Available: https://www.jstor.org/stable/2685209

[53] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: https://doi.org/10.1007/BF00994018

[54] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[55] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *arXiv:1406.1078 [cs, stat]*, Sep. 2014, arXiv: 1406.1078. [Online]. Available: http://arxiv.org/abs/1406.1078

[56] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997, conference Name: IEEE Transactions on Signal Processing.

[57] Y. Freund and R. E. Schapire, "Large Margin Classification Using the Perceptron Algorithm," *Machine Learning*, vol. 37, no. 3, pp. 277–296, Dec. 1999. [Online]. Available: https://doi.org/10.1023/A:1007662407062

[58] "alan-turing-institute/sktime," Jun. 2021, original-date: 2018-11-06T15:08:24Z. [Online]. Available: https://github.com/alan-turing-institute/sktime

[59] H. Deng, G. Runger, E. Tuv, and M. Vladimir, "A Time Series Forest for Classification and Feature Extraction," *arXiv:1302.2277 [cs]*, Feb. 2013, arXiv: 1302.2277. [Online]. Available: http://arxiv.org/abs/1302.2277

[60] J. Lines, S. Taylor, and A. Bagnall, "Time Series Classification with HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles,"

*ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 5, pp. 52:1–52:35, Jul. 2018. [Online]. Available: https://doi.org/10.1145/3182382

[61] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 1578–1585.

[62] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, Jul. 2019, arXiv: 1809.04356. [Online]. Available: http://arxiv.org/abs/1809.04356

## A Papers Submitted and Under Preparation

- Z. Lu, O. Amin, and B. Shihada, ”A Molecular-based Authentication and Authorization for Industrial Internet of Things Systems”, IEEE Internet of Things Journal, 2021.
- Z. Lu and B. Shihada, ”Application of Deep Learning and Federated Learning to Sleep Stage Prediction Task Using Data From Wearable Device”, IEEE Globecom, in preparation, 2022.