

# 修士論文

感染症例数予測のための時系列解析手法及び機械学習の応用と検証

Application and Validation of Time Series Analysis Methods

and Machine Learning for Forecasting the Number of Infection Cases

---

## 報告者

学籍番号: 1255055

氏名: 土肥 直樹

---

## 指導教員

星野 孝総 准教授

---

令和5年2月17日

高知工科大学大学院工学研究科

基盤工学専攻電子・光工学コース

---

# 目次

<b>第1章</b>	<b>序論</b>	<b>1</b>
1.1	研究背景	1
1.2	関連研究	2
1.3	研究目的	3
<b>第2章</b>	<b>実験手法</b>	<b>4</b>
2.1	統計的手法	4
2.1.1	時系列解析と統計量	4
2.1.2	定常性	4
2.1.3	反転可能性	5
2.1.4	Auto regressive Integrated Moving Average (ARIMA)	5
2.1.5	Seasonal ARIMA (SARIMA)	6
2.1.6	Akaike Infomation Criterion (AIC)	7
2.2	深層学習手法	8
2.2.1	Recurrent Neural Network (RNN)	8
2.2.2	Long Short-Time Memory (LSTM)	10
2.2.3	Sequence to Sequence (Seq2Seq)	12
2.2.4	再帰型のニューラルネットワークにおける入出力関係	12
2.2.5	損失関数:Mean Seqyeare Error (MSE)	13
2.2.6	評価関数:Root Mean Squared Error (RMSE)	14
<b>第3章</b>	<b>実験条件</b>	<b>15</b>
3.1	全実験共通の条件	15
3.2	実験1における実験条件	15
3.2.1	データ収集	15
3.2.2	Deep Learning 用入力データのスケーリング	17
3.2.3	データセットの作成	17
3.2.4	モデル構築	17
3.2.5	評価関数によるモデルの評価と比較	18
3.3	実験2における実験条件	19
3.3.1	データ収集	19
3.3.2	Deep Learning 用入力データのスケーリング	19
3.3.3	データセットの作成	20
3.3.4	モデル構築	20
3.3.5	評価関数によるモデルの評価と比較	20
3.4	実験3における実験条件	21
3.4.1	データ収集	21
3.4.2	Deep Learning 用入力データのスケーリング	22
3.4.3	データセットの作成	22
3.4.4	モデル構築	22
3.4.5	評価関数によるモデルの評価と比較	23

---

第4章	結果と考察	24
4.1	COVID-19における統計モデルの探索結果	24
4.2	実験1: COVID-19の第2波までを訓練させたモデルによる予測結果と考察	25
4.2.1	結果	25
4.2.2	考察	28
4.3	実験2: COVID-19の第7波までを訓練させたモデルによる予測結果と考察	32
4.3.1	結果	32
4.3.2	考察	36
4.4	サル痘のデータにおける統計モデルの探索結果	39
4.5	実験3: サル痘のデータを訓練させたモデルの予測結果と考察	39
4.5.1	結果	39
4.5.2	考察	43
第5章	結論	46
	謝辞	48
	参考文献	49
	研究業績	54

# 第1章 序論

## 1.1. 研究背景

紀元前より人類は感染症と闘い続けてきた。主に流行してきたウイルスはインフルエンザ、ペスト、コロナウイルス、天然痘ウイルスである。そして近年では、コロナウイルスとサル痘が形を変えて数度流行している。特に直近では、COVID-19の流行やサル痘が流行している。COVID-19は、ウイルス分類学上 Coronaviridae 属に分類されるウイルスによって引き起こされる。また、サル痘及び天然痘は、Orthopoxvirus 属に属するウイルスによって引き起こされる。

コロナウイルスは、その表面に太陽のコロナの様な突起を持つ [1]。コロナウイルスは、鳥類や豚、牛などの動物と人に感染する。コロナウイルスに感染すると人の場合、呼吸器症状を引き起こす。中でも、2002年11月頃から流行したSARS コロナウイルスや2012年頃から流行したMERS コロナウイルスは人体に対して重症化の恐れがある呼吸器症状を引き起こす。さらに、2019年11月頃から中国の湖北省武漢市が発生源と言われている新型コロナウイルスが流行した。この新型コロナウイルスは、Severe Acute Respiratory Syndrome Coronavirus2 (SARS-CoV-2) と名付けられた。また、SARS-CoV-2によって引き起こされる症状については Coronavirus disease-2019(COVID-19) と名付けられた [2]。Jason ら [3] によると、SARS-CoV-2はSARS コロナウイルスよりも人の細胞に取りつきやすいとされている。そのため、その感染力は強く日本国内に置いても2020年3月頃から流行している。さらに、現在も世界中で流行し続けている。

世界保健機関 (World Health Organization:WHO) の週刊報告書 [4] によると、本稿執筆時時点(2022年12月14日)で全世界で6億4500万以上の確認症例、660万人以上の死者が報告されている。日本国内においても、第112回新型コロナウイルス感染症対策アドバイザリーボード(厚生労働省)[5]によると現在でも病床使用率が5割を上回る地域も多い。COVID-19によって生活様式は大きく変化し、経済活動に対しても大きな影響を与え続けている。

また、Magnus ら [6] によると、初めて天然痘のような症状が観測され症例者が隔離されたのは1958年のことである。サル痘は、Arita ら [7] によるとこれまで1960年から数度に渡りアウトブレイクを繰り返してきている。直近でも、世界各国でサル痘ウイルスが再度流行し、日本の厚生労働省は、各都道府県衛生主管部に情報提供を求めている [8]。

このような背景から、近年では医療体制の崩壊を防ぐことや政治的意思決定が重要になってきている。医療体制の崩壊は、病床や医療関係者の不足が原因で起こる。医療体制の崩壊によって、感染者全員が十分な治療を受けることができずに重症化する患者が発生してしまう。予測により、感染者の療養に必要な病床数や感染者や重症化患者を減らすために必要なワクチンの供給量の把握が可能となる。

また、政治的側面では、感染爆発に伴う緊急事態宣言やGotoキャンペーンの停止など

重要な意思決定が必要になる。これらの判断を間違えると、企業や自治体が大きな経済的損失を被ることになる。その為、症例数を実データから予測することでこれらの政治的意  
思決定を根拠をもとに行うことが可能となる。

従って、公衆衛生、政治的意決定、病床数及びワクチンの確保のために、各種感染症  
の予測モデルの需要が高まってきている。そのため、本研究では COVID-19 及びサル痘  
の症例数予測モデルを構築する。

### 1.2. 関連研究

Wu ら [9] は、疫病予測分野において統計モデルと Neural Network モデルの比較を行っ  
た。彼らは、Autoregressive Integrated Moving Average (ARIMA)[10]、Long Short-Term  
Memory (LSTM)[11]、Attention 機構付きの Sequence to Sequence(Seq2Seq + Attention)、  
Transfor-

mer[12] の予測精度を比較した。彼らの研究では、機械学習モデルと統計モデルの比較  
基準として ARIMA の Root Mean Square Error (RMSE) を設定している。データセット  
は、Centers for Disease Control and Prevention(CDC) 提供のインフルエンザ様疾患発生  
数のデータセットを用いていた。CDC 提供のデータセットから過去 10 週間の特徴量を入  
力値とすることで直近 4 週間のインフルエンザ様疾患の発生数を予測させている。彼らの  
研究結果によると、ARIMA の RMSE から最も誤差を減少できたモデルは Transformer で  
あった。

彼らの研究で最も精度が高かった Transformer[12] は、自然言語処理分野から登場し  
近年様々な領域で応用され高い汎用性と精度を記録している [13, 14, 15, 16, 17]。特に、  
知覚情報処理では Vision Transformer[18] が用いられている。自然言語処理 (NLP) では  
Transformer[12] から更にモデルの改良及び発展を遂げ BERT[19] などの Transformer のモ  
デルアーキテクチャを基本とした多くのモデルが誕生している。本研究で我々が行う時  
系列予測においては、Zhou ら [20] によって時系列予測タスクに適応された Informer が提  
案されている。Informer は、モデルの予測値を次時刻の予測の際の入力として再帰的に  
予測を行うのではなく 1 度の予測で複数時刻を予測している。再帰的に予測する際は、  
累積的に初期の残差が影響してしまう欠点があるため 1 度に複数時刻予測できることは予  
測において利点である。

しかし、Lee らは Transfomer (彼らの論文内では Vision Transfomer) では大規模なデー  
タセットでの学習が必要であり、小規模のデータセットの学習には新たな手法の提案が必  
要だと述べている [21]。我々が行う感染症の時系列予測では、日更新データであり、予測  
が求められる緊急性のある状態では数ヶ月もしくは数年分のデータセットしか存在し  
ない。そこで、本研究では機械学習のモデルとして Transformer よりもモデルサイズの小  
さい Recurrent Neural Network(RNN) と RNN から派生したモデルを構築し予測検証を  
行う。

また、COVID-19 の累積症例数を統計モデルと深層学習モデルで予測した ArunKumar  
ら [22] は、国毎に適切な予測モデルが異なると述べている。これは、国毎に感染対策等が  
異なることに起因する。特に、累積症例数が一定の割合で増加する国の場合、統計モデル  
の方が予測精度が高い。また、波形が非線形的な場合、つまり感染爆発が起きている国で  
は深層学習モデルでの予測が適していると述べている。従って、症例数の時系列予測を行

う本研究においても統計モデルと機械学習モデルの比較を行う。

更に、Spyros ら [23] は、ARIMA 等の統計モデルと RNN や LSTM 等の機械学習の予測精度を比較した。彼らは、他の研究での機械学習を用いた時系列予測の精度に疑問を抱いた。そこで、時系列予測タスク一般に機械学習によって ARIMA 等の従来の統計的な時系列予測手法を代替可能か検証した。彼らは、M3 コンペティションと呼ばれるマーケットの予測を行うコンペティションのデータセットを用いてモデルを構築した。その結果、統計モデルの予測精度が機械学習モデルと比較した際に、モデルのパラメータ数の少なさと比例して高い傾向があった。

前述の Wu ら [9]、ArunKumar ら [22]、Spyros ら [23] の研究より、本研究においても統計モデルと機械学習モデルを構築し比較検証することは重要な意味があると我々は考える。そこで、本研究では Wu ら [9] の研究を参考に ARIMA の RMSE をベースラインとした RMSE の増減率を算出し彼らのインフルエンザ様疾患発生予測モデルの結果と比較を行う。また、Zhou ら [20] の研究を参考に我々の機械学習モデルにおいても複数時刻予測できるモデルを構築した。

### 1.3. 研究目的

本研究では、数多くある感染症に汎用的かつ予測精度の高いモデル構築の為、1.2 節に挙げた理由から以下 3 つの実験を行った。

まず、実験 1 では、以降で行う実験 2 及び実験 3 の為最も良い予測精度を与える RNN における入出力関係を探すため、数値実験として第 2 波までの COVID-19 の症例数のデータを統計モデル (ARIMA、SARIMA) に、症例数、回復者数、死者数のデータを 3 種類の入出力の RNN (3.2.4 節に後述) で学習させた。そして、Root Mean Square Error (RMSE) を用いてテスト区間の予測精度を評価し ARIMA の RMSE を基準として以降の実験で用いるモデルを決定した。

次に、実験 2 では実験 1 で決定した入出力関係 (3 次元入力 3 次元出力の Many to Many) を用いて第 7 波までの COVID-19 の症例数、回復者数、死者数のデータを 4 種類の再帰型のニューラルネットワークで訓練させた。これによって、どのモデルが汎用的に予測可能なモデルであるか検証することを目的とする。

最後に、実験 3 では実験 2 と同じ実験条件でアメリカ、ドイツ、カナダのサル痘の症例数のデータを 4 種類の再帰型のニューラルネットワークで訓練させた。これにより、実験 2 で使用したアルゴリズムが他の感染症でも使用可能か明らかにすることを目的とする。

## 第2章 実験手法

### 2.1. 統計的手法

#### 2.1.1. 時系列解析と統計量

時系列解析は、一般的な統計分析と同じく基本統計量が基本となる。式 2.1 から式 2.4 に時系列解析における各統計量を与える。一般的な統計学と同じく、時刻  $t$  におけるデータ  $y_t$  に対する基本統計量の期待値と分散はそれぞれ、式 2.1 と式 2.2 で与えられる。そして、同一系列における異点間の共分散である自己共分散  $\gamma_{kt}$  は式 2.3 で、自己共分散  $\gamma_{kt}$  が単位に依存しないよう基準化した自己相関係数  $\rho_{kt}$  が式 2.4 で与えられる。

$$\mu_t = E(y_t) \quad (2.1)$$

$$\text{Var}(y_t) = E[(y_t - \mu_t)^2] \quad (2.2)$$

$$\begin{aligned} \gamma_{kt} &= \text{Cov}(y_t, y_{t-k}) \quad (2.3) \\ &= E[(y_t - \mu_t)(y_{t-k} - \mu_{t-k})] \end{aligned}$$

$$\begin{aligned} \rho_{kt} &= \text{Corr}(y_t, y_{t-k}) \quad (2.4) \\ &= \frac{\text{Cov}(y_t, y_{t-k})}{\sqrt{\text{Var}(y_t)\text{Var}(y_{t-k})}} \end{aligned}$$

自己相関係数は、2つの時刻の間にある時刻との関係も求まる。2点間のみの相関は、偏自己相関と呼ばれ、式 2.5 で与えられる。

$$\frac{\text{Cov}(y_t, y_{t-k} | y_{t-1}, y_{t-2}, \dots, y_{t-k+1})}{\sqrt{\text{Var}(y_t | y_{t-1}, y_{t-2}, \dots, y_{t-k+1}) \text{Var}(y_{t-k} | y_{t-1}, y_{t-2}, \dots, y_{t-k+1})}} \quad (2.5)$$

また、自己相関関数及び偏自己相関関数の式において  $k$  の変数の関数としてグラフ上に自己相関係数または偏自己相関係数をプロットしたものをコレログラムと呼ぶ。

#### 2.1.2. 定常性

時系列データでは、確率過程が(弱)定常性を満たすか否かの判断が重要である。

(弱)定常性の定義を式 2.6 と式 2.7 に与える。ただし、 $\mu$  は定数、 $\gamma_k$  は自己共分散。なお、(弱)定常性としたのは、強定常性が存在するからである。しかし、強定常性を満たす時系列は実世界には少ない。その為、以降では弱定常性のことを定常性と記す。定常性

は、データの各時刻における期待値が一定。かつ、自己相関が任意の時刻  $k$  だけ離れた  $y_t$  と  $y_{t-k}$  の相関関係が時点には依存せず、時刻差  $k$  のみに依存することである。

$$E(y_t) = \mu \quad (2.6)$$

$$\begin{aligned} Cov(y_t, y_{t-k}) &= E[(y_t - \mu)(y_{t-k} - \mu)] \\ &= \gamma_k \end{aligned} \quad (2.7)$$

定常性を満たさない確率過程を、非定常過程と呼ぶ。非定常過程には、トレンドが存在し、時刻によって分散が変動する。

後述の Moving Average(MA) 過程は、ホワイトノイズの線形和のため定常過程となる。しかし、Auto Regressive(AR) 過程は、定常と非定常のどちらかをもつ。そのため、後述する ARIMA 過程などでは自己回帰過程が定常の時、その系列は定常性を満たすこととなる。

### 2.1.3. 反転可能性

AR 過程及び MA 過程では、それぞれがお互いを表現できる可能性が存在する。このことを、反転可能性と呼ぶ。MA( $q$ ) 過程が反転可能なとき、AR( $\infty$ ) と表現できる。つまり、無限の項を使用することで MA 過程を AR 過程で表現できる。同様に、定常性を満たす AR( $p$ ) 過程は、反転可能なとき MA( $\infty$ ) と表記できる。つまり、無限に項を使用することで MA 過程により AR 過程を表現できる。

### 2.1.4. Auto regressive Integrated Moving Average (ARIMA)

時刻  $t$  のデータを  $y_t$  とするとき、式 2.8 のように時点のずれを表現するラグ演算子  $L^d$  を用いると、差分系列は式 2.9 のように表せる。従って、 $d$  階差分系列を  $\Delta^d y_t$  (ただし、 $\Delta^0 y_t = y_t$ ) と表記することにする。このとき、AR の次数  $p$ 、MA の次数  $q$ 、期待値  $\mu$ 、定数  $c$ 、過去の時刻  $t-p$  のデータ  $y_{t-p}$ 、過去の時刻  $t-q$  までの誤差項  $\varepsilon_{t-q}$ 、自己回帰係数  $\phi$ 、移動平均係数  $\theta$ 、平均 0 で分散  $\sigma^2$  の正規分布に従う誤差  $\varepsilon_t$  が与えられたとき、ARIMA( $p, d, q$ ) は (2.10) 式のように定式化される。なお以降では、ARIMA( $p, d, q$ ) を ARIMA と表記し、使用している変数は、2.1.5 節においても使用する。

$$L^d y_t = y_{t-d} \quad (2.8)$$

$$\Delta^d y_t = (1 - L^d) y_t \quad (2.9)$$

$$\left(1 - \sum_{i=1}^p \phi_i L^i\right) \Delta^d y_t = \left(1 + \sum_{j=1}^q \theta_j L^j\right) \varepsilon_t \quad (2.10)$$

ARIMA は、Auto Regressive (AR) 過程と Moving Average(MA) 過程の 2 つの要素から成り立つ。

$p$  次の AR モデル AR( $p$ ) は式 2.11 によって定式化される。



$$y_t = c + \sum_{i=1}^p \phi_i y_{t-i} + \varepsilon_t \quad (2.11)$$

定常的な AR モデルは、定数  $c$  の周辺で一定の分散に従って繰り返すモデルであり、そのような時系列データをモデル化することに有用である。また、 $y_t \sim \text{AR}(p)$  ( $y_t$  が  $\text{AR}(p)$  過程に従う) の場合、2.1.1 節で述べたコレログラム (縦軸は自己相関係数または偏自己相関係数、横軸はラグ) においてその自己相関関数は指数的に減衰し、偏自己相関関数は  $p+1$  次以降 0 となる。

$q$  次の MA モデル  $\text{MA}(q)$  は式 2.12 によって定式化される。

$$y_t = \mu + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t \quad (2.12)$$

MA モデルは、期待値  $\mu$  の周辺で一定の分散に従って繰り返すモデルであり、そのような時系列データをモデル化し表現することに対して有用である。

$y_t$  が  $\text{MA}(q)$  過程に従う場合、2.1.1 節で述べたコレログラム (縦軸は自己相関係数または偏自己相関係数、横軸はラグ) においてその偏自己相関関数は指数的に減衰し、自己相関関数は  $q+1$  次以降 0 となる。

ARIMA モデルが反転可能である条件は、ARIMA モデルを構成する MA モデルが反転可能であること。また、ARMA モデルが定常である条件は、ARIMA モデルを構成する AR モデルが定常であることである。

### 2.1.5. Seasonal ARIMA (SARIMA)

Seasonal AutoRegressive Integrated Moving Average (SARIMA) は、季節変動成分に対しても ARIMA を適用したモデルである。季節変動成分とは、日単位データの場合 1 週間周期、月単位データなら 1 年周期で変動する成分のことである。

図 2.1 に、1 週間の季節変動成分の捉え方を与える。図 2.1 中の  $y_t$  は日単位データである。ARIMA モデルでは、緑色で囲われた部分に対して相関を持つようにモデルを構築する。それに加えて SARIMA モデルは赤色で囲われた季節変動の周期成分に対しても相関を持つようにモデルを構築する。これによって、長期的な成分をモデルに含むことが可能になる。

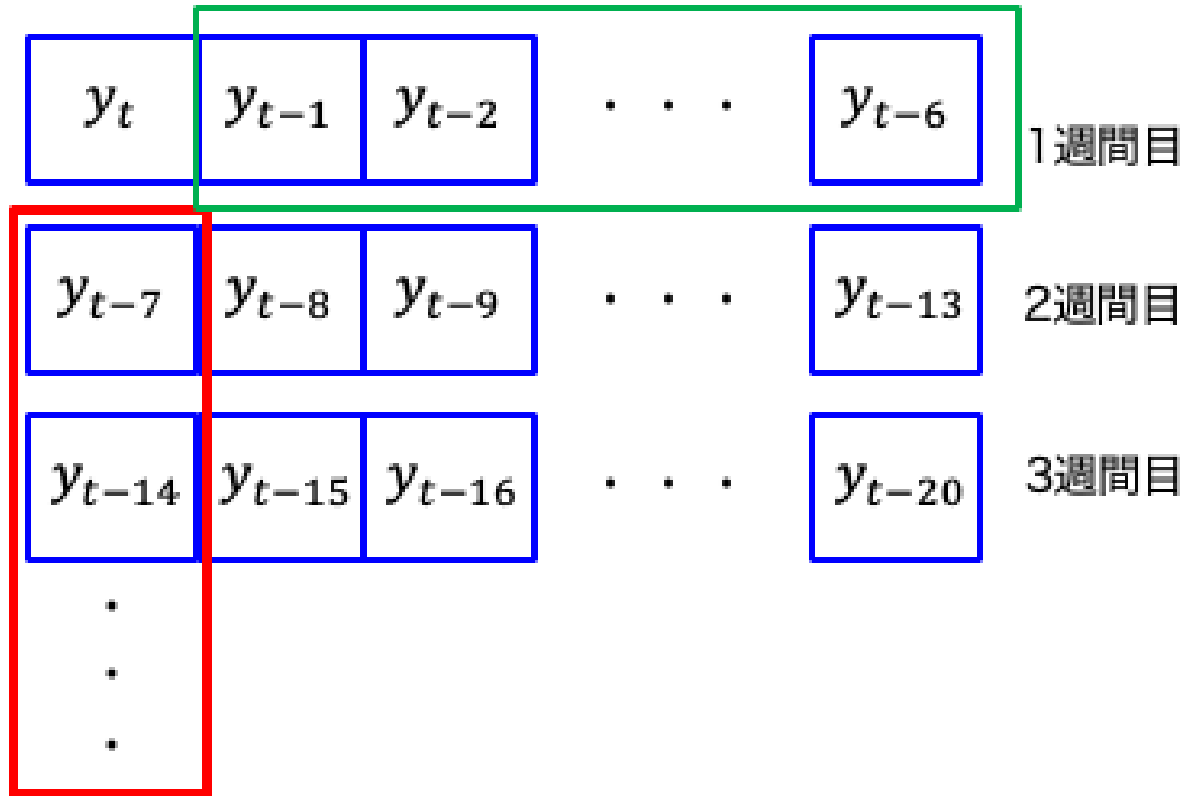


図 2.1: 季節変動成分の捉え方

ARIMA と同様に、季節変動成分に対するラグ演算子を  $L^{sD}$  とし、 $D$  階季節差分をとることを  $\Delta_s^D y_t$  (ただし、 $\Delta_s^0 y_t = y_t$ ) と表記することにする。そして、季節自己回帰の次数  $P$ 、季節移動平均の次数  $Q$ 、季節自己回帰係数  $\Phi$ 、季節移動平均係数  $\Theta$  が与えられたとき、SARIMA( $p, d, q, P, D, Q$ ) は、式 2.13 で定式化される。

$$\left(1 - \sum_{i=1}^p \phi_i L^i\right) \left(1 - \sum_{l=1}^P \Phi_l L^{sl}\right) \Delta^d \Delta_s^D y_t = \left(1 + \sum_{j=1}^q \theta_j L^j\right) \left(1 + \sum_{J=1}^Q \Theta_J L^{sJ}\right) \varepsilon_t \quad (2.13)$$

### 2.1.6. Akaike Infomation Criterion (AIC)

Akaike Infomation Criterion は、モデルのデータに対する適応度とパラメータに起因する複雑さの兼ね合いを評価する指標である。AIC は、2.14 式で与えられる。ここで、 $L$  は対数尤度、 $k$  はモデルに含まれるパラメータ数を表す。尤度は、モデルのデータへの適応度を表現する。尤度は、対数をとった方が扱いやすい為対数尤度を使用する。

$$AIC = -2 \ln(L) + 2k \quad (2.14)$$

第 1 項目の  $-2 \ln(L)$  は、尤度の増加に対して単調減少するが減少幅は次第に小さくなる。また、第 2 項目の  $2k$  はパラメータ数  $k$  に比例して増加し、パラメータ数の多さに対する罰則項として扱われる。従って、AIC は同じモデルの適応度である場合、パラメータ

数が少ないモデルの方が小さい AIC となる。通常、モデルのパラメータ数が多いモデルよりも少ないモデルで予測精度が高い方が良いとされる。なぜならば、パラメータ数が多いモデルは過学習しやすく適応度が高い場合も、テストデータに対して同じ精度を発揮することができないからである。また、説明性の観点からもパラメータは少ない方が良い。本稿では、入力変数の組み合わせを変更し AIC が最小となる入力変数の組み合わせを総当たりに探索し推定することを最小 AIC 推定と呼ぶ。

ここで、 $i$  番目のサンプルに対する回帰モデルの誤差  $\epsilon_i$  が平均 0 かつ分散  $\sigma^2$  の正規分布  $N(0, \sigma^2)$  に従うと仮定する。また、サンプル数  $N$ 、パラメータ  $\theta$ 、観測値  $y$ 、モデルによる予測値  $\hat{y}$ 、 $RSS = \sum_{i=1}^N (y_i - \hat{y}_i)^2$  を与える。

このとき、対数尤度関数は 2.15 式で定式化される。

$$\log L(\theta) = -\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \hat{y}_i)^2 - \frac{N}{2} \log(2\pi\sigma^2) \quad (2.15)$$

$$= -\frac{RSS}{2\sigma^2} - \frac{N}{2} \log(2\pi\sigma^2) \quad (2.16)$$

ここで、 $\sigma^2$  で微分し、極点を求めると、

$$\begin{aligned} \frac{\partial \log L(\theta)}{\partial \sigma^2} &= \frac{RSS}{2\sigma^4} - \frac{N}{2\sigma^2} \\ &= 0 \end{aligned} \quad (2.17)$$

となる。

従って、最尤推定量  $\sigma^2$  は  $\hat{\sigma} = \frac{RSS}{N}$  となる。このとき、最大対数尤度は 2.18 式で与えられる。

$$\ln L = -\frac{N}{2} - \frac{N}{2} \ln\left(\frac{2\pi RSS}{N}\right) \quad (2.18)$$

従って、AIC は回帰モデルの誤差  $\epsilon_i$  が平均 0 かつ分散  $\sigma^2$  の正規分布  $N(0, \sigma^2)$  に従うと仮定した時 2.19 式で与えられる。

$$AIC = 2k + N \ln \frac{(RSS)}{N} \quad (2.19)$$

## 2.2. 深層学習手法

### 2.2.1. Recurrent Neural Network (RNN)

Rosenblatt[24] は、人間の脳内のニューロンを数式で表現したパーセプトロンを考案した。パーセプトロンをつみ重ねた多層パーセプトロンは、パターン認識などの様々な課題を解決した。しかし、多層パーセプトロンでは、時系列データのパターンを十分に学習する

ことが不可能だった。この問題を解決したのが Recurrent Neural Network(RNN)[25, 26] である。

RNN の中間層 (隠れ層) は、特別にリカレント層と呼ばれる。ここで、時刻  $t$  における入力  $x_t$ 、入力  $x_t$  に対する重み  $W_x$ 、1つ前の時刻  $t-1$  における出力  $h_{t-1}$  に対する重み  $W_h$ 、バイアス  $b$  が与えられた時、リカレント層の時刻  $t$  における出力  $h_t$  は式 2.20 で定式化される。この時、リカレント層の出力  $h_t$  は隠れ状態ベクトルと呼ばれ、時刻  $t$  における情報を保持している。

$$h_t = \tanh(x_t W_x + h_{t-1} W_h + b) \quad (2.20)$$

式 2.20 では、行列の積とそれらの和を計算し、 $\tanh$  関数によって時刻  $t$  における出力  $h_t$  を算出する。 $\tanh$  では、情報の重要度によって  $-1$  から  $+1$  の値に写像される。

以下図 2.2 に、RNN のリカレント層の構造を示す。なお、図 2.2 中の  $t$  は時刻、 $x_t$  は入力、 $h_t$  は隠れ状態ベクトルを意味する。

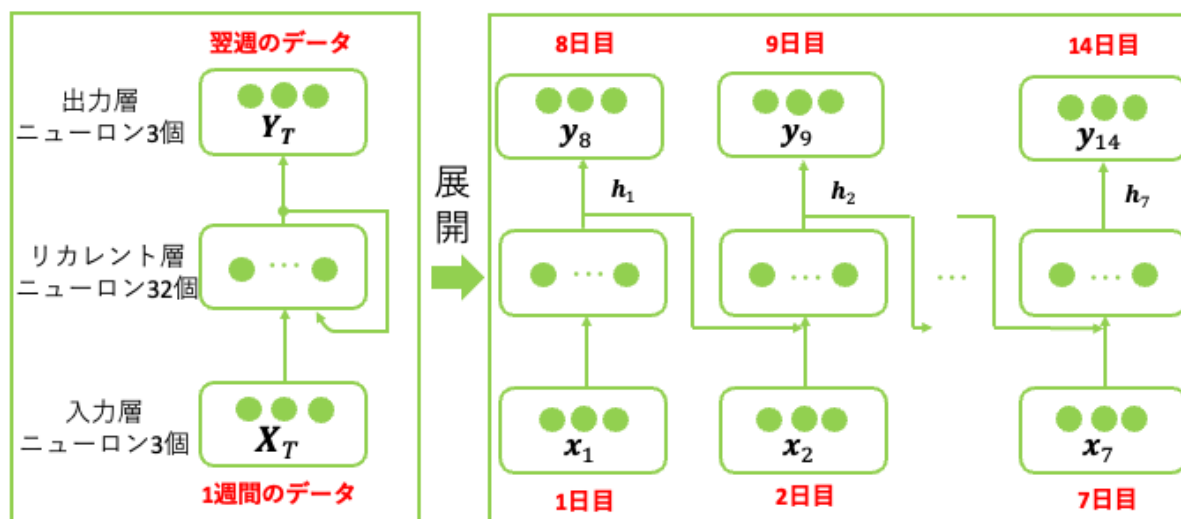


図 2.2: Recurrent Neural Network

リカレント層は、入力  $x_t$  に対応して隠れ状態ベクトル  $h_t$  が出力される。隠れ状態ベクトル  $h_t$  は、過去の記憶を表現している。展開された RNN の形は、多層パーセプトロンと変わらないがすべて同じリカレント層に存在する点で異なる。図 2.2 のようにそれぞれの時刻において、リカレント層は時刻  $t$  における入力  $x_t$  および一つ前の時刻  $t-1$  における出力  $h_{t-1}$  を入力として用いる。

それら 2つをリカレント層の入力として用いて計算することでその時刻における出力が計算される。この様に回帰的に計算することで、過去の情報を保持したまま新たな入力を計算することが可能になった。

展開された RNN は、通常のニューラルネットワークと同様に Backpropagation[27] により勾配を逆伝播することで学習することが可能である。ただし、時系列データにおける誤差逆伝搬法は時間軸方向に展開した Backpropagation が行われる [28]。その為、時間軸方向に展開した Backpropagation の意味で Backpropagation Through Time (BPTT) と呼ばれる [29]。

なお、RNNは後述の Long Short-Time Memory (LSTM)、Sequence to Sequence (Seq2Seq)、Stacked LSTM は概念として RNN に含まれる。そこで以降では、モデルの RNN と概念を区別するため RNN、LSTM、Seq2Seq、Stacked LSTM を含む概念を再帰型のニューラルネットワークとし、モデルを RNN と記述する。

## 2.2.2. Long Short-Time Memory (LSTM)

Long Short-Time Memory(LSTM) は、RNN から派生してできたアルゴリズムある。その為、LSTM も RNN と同じく過去の時刻の中間層の出力を現在の時刻の入力に用いることで時系列データを学習する。

2.2.1 節で述べた、RNN には課題があった。それは、勾配消失問題である [30]。勾配消失問題とは、勾配の逆伝播の際に勾配が層を渡るにつれて小さくなりなくなってしまう問題である [31]。勾配消失問題の原因は、RNN の活性化関数の  $\tanh$  関数にある。 $\tanh$  関数は、微分すると  $\frac{dy}{dt} = 1 - y^2$  になる。 $\frac{dy}{dt} = 1 - y^2$  は、 $x = 0$  の時  $y = 1$  となり、 $x = 0$  から遠ざかるにつれてその値は小さくなる。従って、逆伝播の際に  $\tanh$  関数を通過する度に RNN の勾配は小さくなってしまう。そこで、この問題を解決するために Hochreiter と Schmidhuber が LSTM を提唱した [11, 32]。

LSTM は、人間の脳が持つ長期記憶と短期記憶を参考にしており、長期的な時系列情報と短期的な時系列情報を保持する。これにより、長期的なトレンドと短期的な周期を捉えることが可能になった。LSTM の全体構造は、多層パーセプトロンにおける隠れ層の単純なニューロンを図 2.3 の Long Short-Time Memory ブロックに置き換えた構造となる。

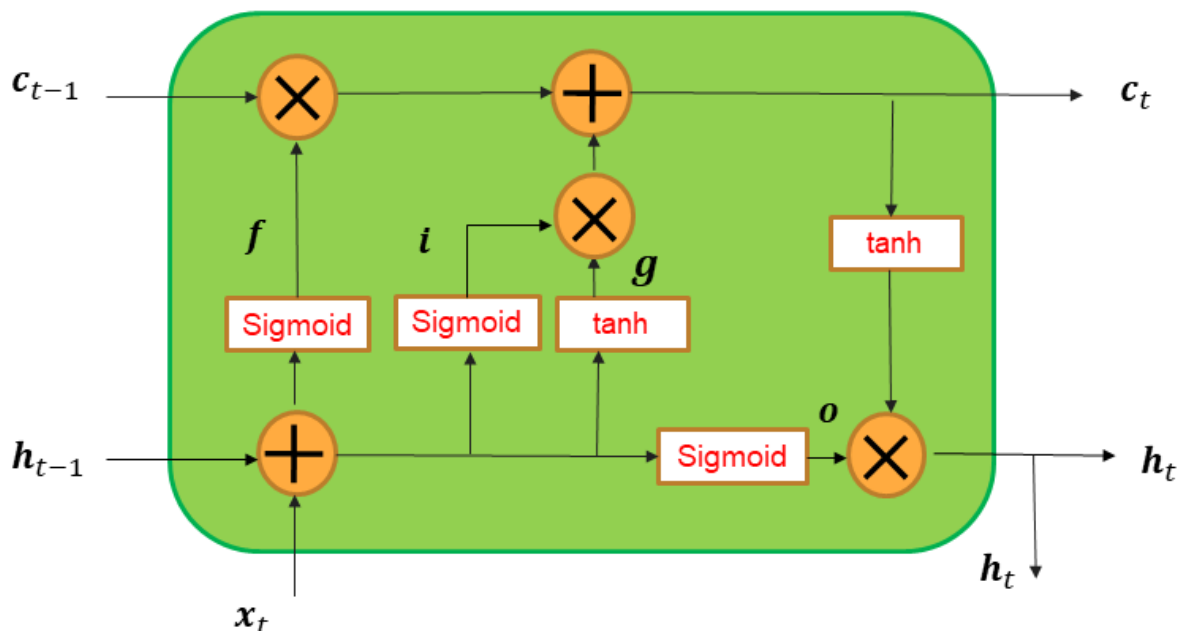


図 2.3: Long-Short Time Memory ブロックの構造

時刻  $t$  における入力  $x_t$ 、入力  $x_t$  に対する重み  $W$ 、1つ前の時刻  $t-1$  における出力  $h_{t-1}$  に対する重み  $U$ 、バイアス  $b$  が与えられた時、記憶セル  $C_t$ 、リカレント層の時刻  $t$  における出力  $h_t$  は式 2.21 から式 2.26 で与えられる。なお、同じ形式の式であっても後述の

ゲートが異なる。そのため、式 2.21 から式 2.24 中の重み及びバイアスには各ゲートを意味する添字を用いた。

$$\mathbf{f} = \sigma(\mathbf{x}_t \mathbf{W}_f + \mathbf{h}_{t-1} \mathbf{U}_f + \mathbf{b}_f) \quad (2.21)$$

$$\mathbf{g} = \tanh(\mathbf{x}_t \mathbf{W}_g + \mathbf{h}_{t-1} \mathbf{U}_g + \mathbf{b}_g) \quad (2.22)$$

$$\mathbf{i} = \sigma(\mathbf{x}_t \mathbf{W}_i + \mathbf{h}_{t-1} \mathbf{U}_i + \mathbf{b}_i) \quad (2.23)$$

$$\mathbf{o} = \sigma(\mathbf{x}_t \mathbf{W}_o + \mathbf{h}_{t-1} \mathbf{U}_o + \mathbf{b}_o) \quad (2.24)$$

$$\mathbf{c}_t = \mathbf{f} \odot \mathbf{c}_{t-1} + \mathbf{g} \odot \mathbf{i} \quad (2.25)$$

$$\mathbf{h}_t = \mathbf{o} \odot \tanh(\mathbf{c}_t) \quad (2.26)$$

LSTM ブロックは、主に4つの要素で構成されている。それは記憶セル  $\mathbf{C}_t$  と3つのゲートと呼ばれるニューロンからなる構成要素である。

まず、入力ゲートである。入力ゲートでは、入力  $\mathbf{x}_t$  及び1つ前の時刻の隠れ状態ベクトル  $\mathbf{h}_{t-1}$  をどの程度反映するか調節する。具体的には、入力ゲートの式 2.23 の Sigmoid 関数によって計算された0から1の値を式 2.22 によって計算された値に掛け合わせる事で新規の情報に対する重み付けを行っている。これにより、現在の時刻  $t$  に入力された新規の情報をどの程度後述する記憶セル  $\mathbf{C}_t$  に反映するか決定される。この時、重み付けの値が1であれば全ての情報が記憶セル  $\mathbf{C}_t$  に反映される。

次に、記憶セル  $\mathbf{C}_t$  である。記憶セル  $\mathbf{C}_t$  は、LSTM において最も重要な構成要素である。記憶セル  $\mathbf{C}_t$  には、時刻  $t$  までの過去の情報が格納される。したがって、記憶セル  $\mathbf{C}_t$  には学習過程で必要と判断されたトレンド成分が格納されている。具体的には、式 2.22 によって新たに追加される情報が計算され1つ前の時刻の記憶セル  $\mathbf{C}_{t-1}$  に追加される。この際、前述の入力ゲートと異なり Sigmoid 関数ではなく tanh 関数が用いられている。これは、ゲートに重み付けを行う機能では無く、新しい情報を追加する機能を持たせたいからである。

次に、忘却ゲートである。忘却ゲートでは、記憶セル  $\mathbf{C}_t$  の内容をどの程度保持するか調節される。入力  $\mathbf{x}_t$  と1つ前の時刻の隠れ状態ベクトル  $\mathbf{h}_{t-1}$  を式 2.21 のように Sigmoid 関数に入力し式 2.25 記憶セル内部の過去の記憶  $\mathbf{C}_{t-1}$  と掛け合わせる。Sigmoid 関数の出力は0から1の間である。従って、過去の記憶をどの程度保持するか調整することが可能である。このとき、0は完全に忘却することを意味し、1は全ての情報を保持することを意味する。

最後に、出力ゲートである。

現在の時刻  $t$  における隠れ状態ベクトル  $\mathbf{h}_t$  が与えられたとき、出力ゲートによる出力は式 2.26 によって定式化される。出力ゲートは、情報に対する重み  $\mathbf{o}$  と記憶セル  $\mathbf{C}_t$  によって構成されている。ここで、記憶セル  $\mathbf{C}_t$  を入力とする活性化関数には tanh 関数が用いられている。これは、記憶セル  $\mathbf{C}_t$  の情報に対する強度を-1から1で表現する為に用いられている。

また、 $\mathbf{o}$  は、式 2.24 によって定式化される。 $\mathbf{o}$  は、入力  $\mathbf{x}_t$  と1つ前の時刻の隠れ状態ベクトル  $\mathbf{h}_{t-1}$  を Sigmoid 関数に入力することで算出される。これにより、現在の時刻  $t$  の情報が次の時刻  $t+1$  の隠れ状態ベクトルとしてどれくらい重要か重み付けが行われ、新たな隠れ状態ベクトル  $\mathbf{h}_t$  が出力される。

LSTM において、一番重要な要素は図 2.3 の一番上の線部である。RNN では、逆伝播の際に同じ重みを用いて行列の積演算を繰り返し行われる。そのため、勾配爆発や勾配消失が発生していた。しかし、一番上の線上ある LSTM の演算は、加算と乗算のみである。従って、LSTM の逆伝播では、要素ごとの乗算を行うだけの演算になる。その為、逆伝播の際に勾配を劣化させることなく伝播することが可能となり勾配爆発や勾配消失が起きにくくなる。

なお、Maaliw[33] らの COVID-19 予測モデルでは、LSTM 層を多層化した Stacked LSTM が用いられており良い予測精度を得ていた。そこで、本研究においてもリカレント層を 2 層用いた Stacked LSTM を作成しモデル比較に用いた。

### 2.2.3. Sequence to Sequence (Seq2Seq)

Sequence to Sequence (Seq2Seq) は自然言語処理の領域で Google 翻訳等にも用いられているモデルである [34]。

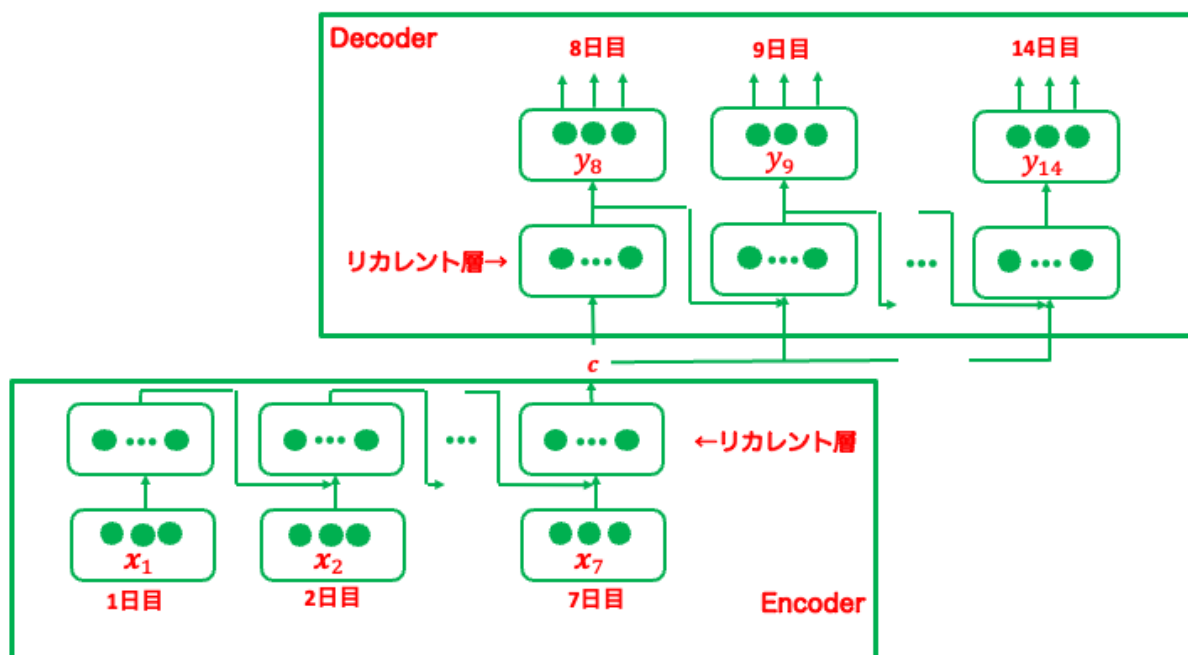


図 2.4: Seq2Seq

Seq2Seq では、Encoder のリカレント層の最終時刻から出力された文脈ベクトル  $c$  を用いて Decoder によって予測を行う。今回は、1 週間の感染状況に関するデータを要約し、要約値を Decoder の入力とすることで翌週を予測するように学習する。本研究では、文脈ベクトル  $c$  が予測にどのような影響を与えるかを観測した。

### 2.2.4. 再帰型のニューラルネットワークにおける入出力関係

RNN は、多種多様な入出力関係を構築することが可能である。RNN が持つ入出力関係は、主に 4 種類ある。それは、One to Many、One to One、Many to One、Many to

Manyである。本研究では Many To Many を用いた。One to One は時系列予測を行わない通常のニューラルネットワークであり、One to Many は時系列予測としての利点がないことから今回は使用しなかった。

ここで、Many To One のデータセットの例を図2.5に与える。この例では、7日間の入力に対して8日目の予測を行っている。通常、再帰型のニューラルネットワークを用いた時系列予測では Abbasimehr[35] らの先行研究で行われているように Many to One で行われることが多い。しかし、Many to One には大きな欠点がある。それは、予測値と観測値との残差が後の予測に影響を与えることである。これは、Many to One では1つ先の時刻の値しか出力しないため複数時刻先を予測するには、予測値を次の時刻を予測するため再起的に入力する必要があることに起因する。そのため、本研究では使用しなかった。

そこで、本研究では図2.6に示す1度の予測で複数時刻を予測可能な Many to Many の再帰型のニューラルネットワークを構築した。

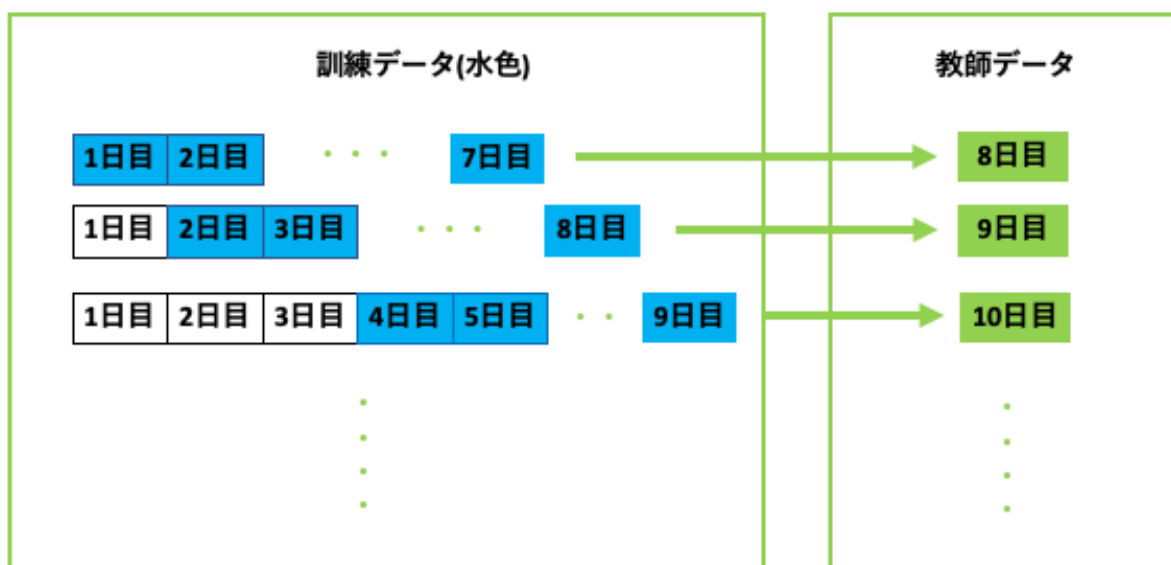


図 2.5: Many To One のデータセット

### 2.2.5. 損失関数:Mean Sqeyare Error (MSE)

本研究では、損失関数に Mean Sqeyare Error (MSE) を使用した。MSE は、観測値を  $y_i (i = 1, 2, 3, \dots, n)$ 、モデルより計算した予測値を  $\hat{y}_i$  としたとき観測値と予測値が近づくほど、MSE は0に近づき小さくなる。MSE は、誤差を2乗するため、外れ値の影響を受けやすい。その為、外れ値が予測値に含まれた場合、MSE は急速に大きくなる。

MSE は次の式 2.27 で定式化される。予測値  $\hat{y}_i$  と観測値  $y_i$  の差は2乗されるため必ず正値になる。

$$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.27)$$

また、単層の RNN と単層の LSTM において過学習の傾向が観測された為、式 (2.28) のように L2 ノルムを罰則項として通常の損失関数  $E$  に加え、最終的な損失関数を  $E_w$  とした。なお、 $\lambda$  は正則化を反映するためのパラメータであり  $\lambda = 0.001$  とした。



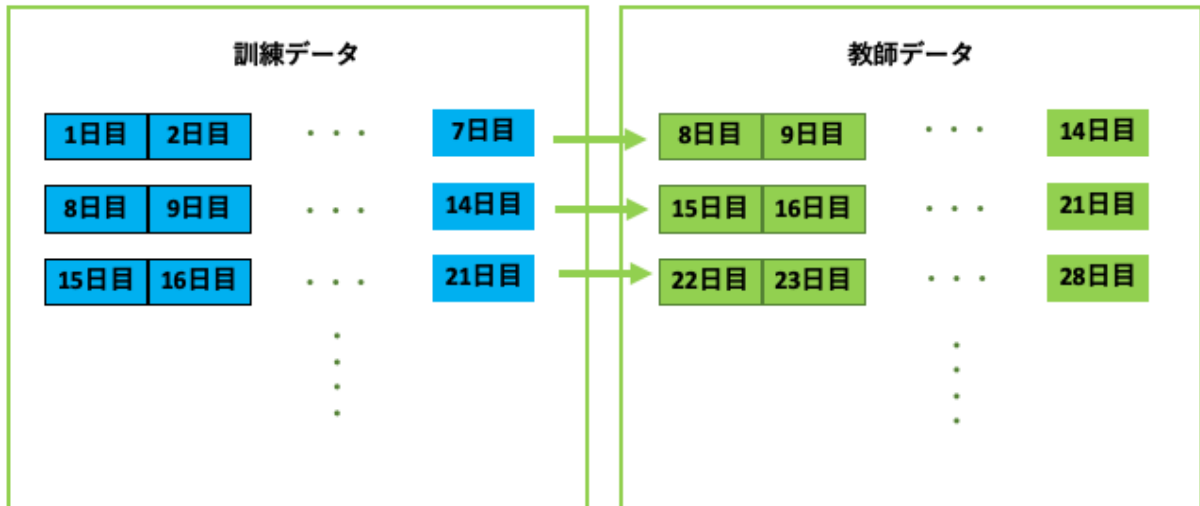


図 2.6: Many To Many のデータセット

$$E_w = E + \lambda \sum_{k=1}^n w_k^2 \quad (2.28)$$

### 2.2.6. 評価関数:Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) は、観測値を  $y_i (i = 1, 2, 3, \dots, n)$ 、モデルの予測値を  $\hat{y}_i$  で与えられたとき、式 2.29 で定式化される。RMSE は MSE 同様、誤差が 2 乗される為外れ値が含まれた場合 RMSE は急速に大きくなる。本研究では、RMSE をモデルの評価関数として用いた。これは、RMSE が MSE のルートを取ることで 2 乗された単位から人間にとって分かりやすい単位になるためである。

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.29)$$

# 第3章 実験条件

## 3.1. 全実験共通の条件

本章では、各実験の実験条件について述べる。共通の条件として、実験の実行、モデルの実装において、プログラミング言語 Python を使用した。時系列解析及び統計モデルの構築には、Python の統計用パッケージの Statsmodels[36] を使用した。機械学習モデルの構築には、オープンソースの Deep Learning 用フレームワークの Keras[37] を使用し、そのバックエンドには Tensorflow[38] を使用した。

本稿で述べる実験は全て以下の手順で行われた。

1. データ収集
2. Deep Learning 用入力データのスケールリング
3. データセット作成
4. モデル構築
5. モデルの評価と比較

本研究では、モデル評価のベースラインとして ARIMA を設定した。ARIMA は、時刻  $t$  において過去の時系列やホワイトノイズと何項分回帰するかを決定する次数  $p$  及び  $q$ 、いくつ階差を取るかを決定する次数  $I$  が存在する。 $p$  が AR の次数、 $q$  が MA の次数、 $d$  が  $I$  の次数と定義される。次数の決定方法は、主に 2 種類ある。1 つは、時刻同士の相関を ACF 及び PACF で算出しプロットすることで決定する方法。もう 1 つは、ある次数範囲におけるすべてのモデルを構築し、最も AIC が低いモデルを総当たりの選択する方法である。通常次数選択において、ARIMA は ACF や PACF によって次数を決定することはモデルの柔軟さなどの理由から困難である。Ozaki[39] の研究は、この困難を最小 AIC 推定により取り除ける可能性を数値研究により示している。そこで、本研究では表 3.1 に示した範囲で総当たりの次数探索を行った。そして、各モデルの AIC を算出し、COVID-19 の第 2 波までと第 7 波まで、及びサル痘、それぞれの感染状況のデータにおける  $ARIMA(p, d, q)$  及び  $SARIMA(p, d, q, P, D, Q)_7$  を選択した。

## 3.2. 実験 1 における実験条件

### 3.2.1. データ収集

実験 1 で使用した第 2 波までの COVID-19 に関するデータセットは、米 Johns Hopkins University (JHU) 提供のデータセット [40] を元に編集・使用した。図 3.1 に使用した JHU 提供のデータセットを示す。

表 3.1: 統計モデルの最小 AIC 推定時の次数探索範囲

(a) ARIMA( $p, d, q$ ) の次数探索範囲

	p	d	q
min	1	0	0
max	3	1	3

(b) SARIMA( $p, d, q, P, D, Q$ )<sub>7</sub> の次数探索範囲

	p	d	q	P	D	Q
min	1	0	0	0	0	0
max	3	1	3	3	2	3

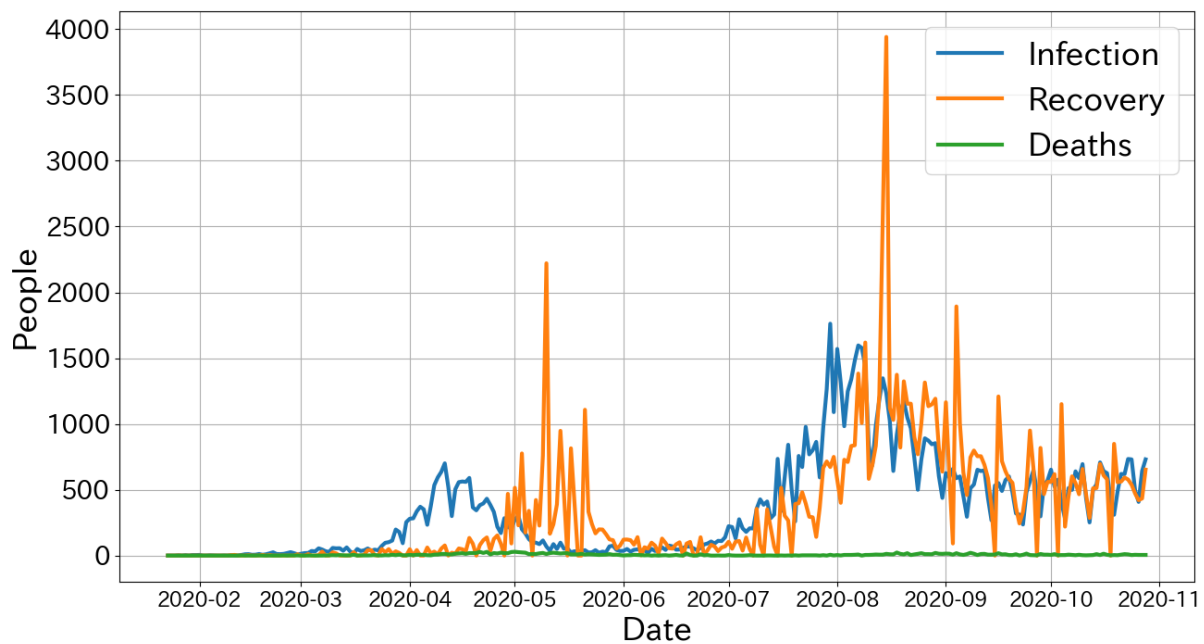


図 3.1: JHU 提供の COVID-19 に関するデータセット

JHU は、各国が公開している COVID-19 に関するデータを集計している。JHU 提供のデータセットには、各国における累積の確認症例数、回復者数、死者数が含まれている。Shahid ら [41] が行った研究では、各国の感染状況の累積データと LSTM 等の機械学習アルゴリズムを用いてモデルを構築していた。しかし、累積データを用いたモデルの予測では予測が 1 日でも大きく外れた場合、後の予測に累積的に影響がある。そこで実験 1 では、JHU 提供のデータセットを  $y_t - y_{t-1}$  のように前日との階差を取ることで 1 日における新規の症例数、回復者数、死者数のデータセットを作成し使用した。これにより、予測が外れた際の累積的な影響を取り除くことが可能となった。また、感染症数理疫学で天然痘の感染予測モデルとして開発されたモデルに SIR (Susceptible-Infective-Removed) [42] がある。SIR は感染症数理疫学で幅広く利用されている。SIR では、数式を利用してある集団における感染状況をモデル化するために感染者数、回復者数、死者数の 3 変数を含んでいる。そこで、実験 1 及び実験 2 においても機械学習を用いて感染状況をモデル化するため 1 日における新規の症例数、回復者数、死者数の 3 変数を用いて RNN のモデル構築を行った。

訓練データは、2020 年 1 月 23 日～同年 10 月 21 日の期間のデータを用い、テストデータは、2020 年 10 月 22 日～同年 10 月 28 日の 1 週間のデータを用いた。なお、RNN の学習においては 3.2.3 節で述べるデータセットの作成時に、7 日間の対応関係が作成できな

かった時系列後半のデータを使用せず、さらにそこから後半の1割を検証データとして使用した。

### 3.2.2. Deep Learning 用入力データのスケーリング

次に、再帰型のニューラルネットワーク用にデータのスケーリングを行った。スケーリングは、式3.1によって行われる。 $x_{min}$  は全入力変数の最小値、 $x_{max}$  は全入力変数の最大値として与えられる。実験1において、ベクトル入力の場合は回復者数の最大値3,941人、スカラー入力の場合は感染者数の最大値1762人とした。最小値は、どちらの場合も0人であった。 $max$  と  $min$  は、それぞれスケーリングの最大値と最小値を意味する。今回は、RNNの中間層の活性化関数である  $\tanh$  関数の出力値に合わせ、 $-1 \sim +1$  に設定した。

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}(max - min) + min \quad (3.1)$$

### 3.2.3. データセットの作成

再帰型のニューラルネットワークは、様々な入出力関係を持つことが可能である。例として、Many To One、Many To Many、One To One、One To Many が存在する。今回は、Many To Many を用いてモデルを構築した。Many To Many 用のデータセットは、1週間分の訓練データに対して次の1週間を教師データとし、図2.6のように作成した。通常、時系列予測タスクにおけるRNNでは、Abbasimehr[35]らの研究で行われているようにMany To Oneの入出力関係で行われることが多い。しかし、この場合出力が1時刻のため、単純に複数時刻先まで予測することができない。複数時刻先まで予測するためには、前時刻の予測出力を再帰的に次の時刻の入力とする必要がある。しかし、この方法では予測に誤差が存在する場合、累積的にその誤差が影響を生むことになる。そこで、本研究では Many To Many の入出力関係を採用しモデルを構築した。

### 3.2.4. モデル構築

統計モデルのARIMA及びSARIMAは、3.1節で述べたように表3.1に示した範囲で最小AIC推定を行い次数を決定しモデルを構築した。その結果は、4.1節で述べる。

実験1のRNNでは、図3.2から図3.4に示す3種類の入出力関係を持つRNNを構築した。

図3.2は、1次元入力1次元出力のRNNである。このモデルには、直近1週間の新規症例数を変数と入力し翌週の新規症例数を出力させる。1週間内の系列特徴を訓練させる事で翌週の症例数数を予測することを目的としている。図3.3は、3次元入力3次元出力のRNNである。このモデルには、直近1週間の新規症例数、新規回復者数、新規死者数を入力し、翌週の新規症例数、新規回復者数、新規死者数を出力させる。図3.2に示した1次元入力1次元出力のRNNよりも多くの変数を使用することで、より精度の高い予測ができると期待される。図3.4は、3次元入力1次元出力のRNNである。このモデルには、直近1週間の新規症例数、新規回復者数、新規死者数を入力し、翌週の新規症例数を出力

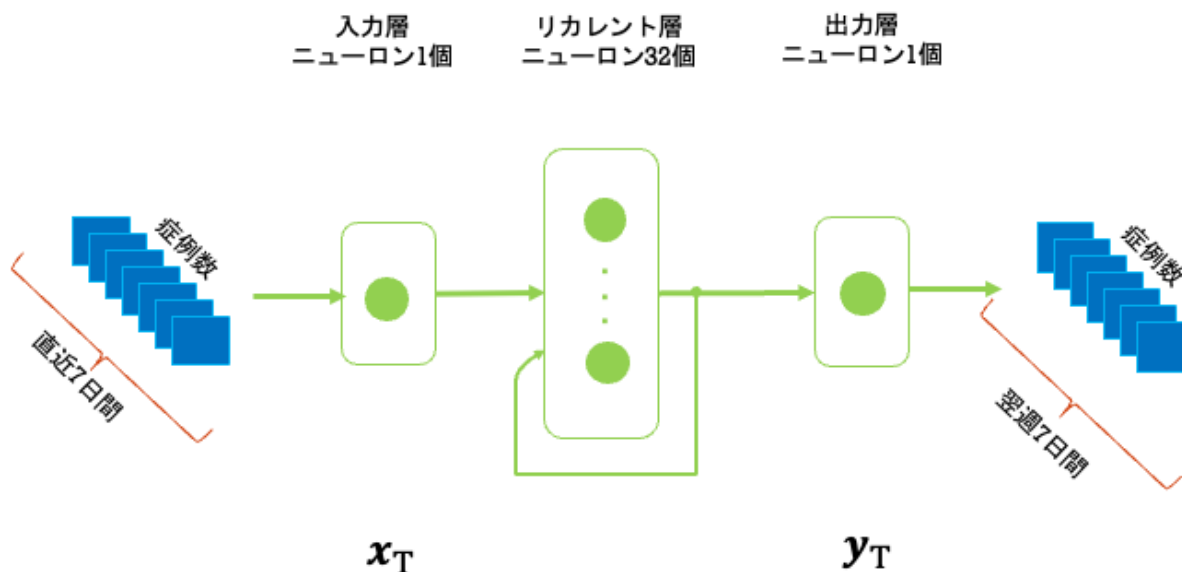


図 3.2: 構築した 1 次元入力 1 次元出力の RNN

させる。このモデルでは、3 変数の入力系列特徴量を 1 次元の系列に写像することによって図 3.2 に示した 1 次元入力 1 次元出力の RNN や図 3.3 に示した 3 次元入力 3 次元出力の RNN よりも精度の高いモデルの構築が期待される。

これら 3 種類の入出力の RNN のパラメータの詳細を表 3.2 に示す。なお、表中の 1 to 1、3 to 1、3 to 3 は、それぞれ 1 次元入力 1 次元出力、3 次元入力 1 次元出力、3 次元入力 3 次元出力を意味する。入出力以外の条件は全て同じ条件に設定し実験した。

表 3.2: 実験 1 における RNN の実験条件

パラメータ 入出力	入力層 (個)	中間層 (個)	出力層 (個)	最適化	エポック数
1 to 1	1	32	1	Adam	100
3 to 1	3	32	1	Adam	100
3 to 3	3	32	3	Adam	100

### 3.2.5. 評価関数によるモデルの評価と比較

各モデルのテスト区間における予測評価を RMSE を用いて行なった。また、Wu ら [9] は、自然言語から始まり、様々な研究分野で最新で高精度な AI を誕生させている Transformer、LSTM、Attention 機構を用いた Seq2Seq を古典的な予測モデルの ARIMA を基準として予測精度の比較を行なった。Wu らの研究では、ARIMA を用いてテスト区間の予測を行い RMSE で評価していた。さらに、Transformer やその他のモデルとの性能差を測る為に、式 3.2 で ARIMA の RMSE を基準としてその誤差の差を計測していた。そこで、本研究においても式 3.2 を用いて ARIMA の RMSE からの性能差を算出した。

$$\left( \frac{RMSE}{ARIMA's RMSE} - 1 \right) \times 100 \tag{3.2}$$

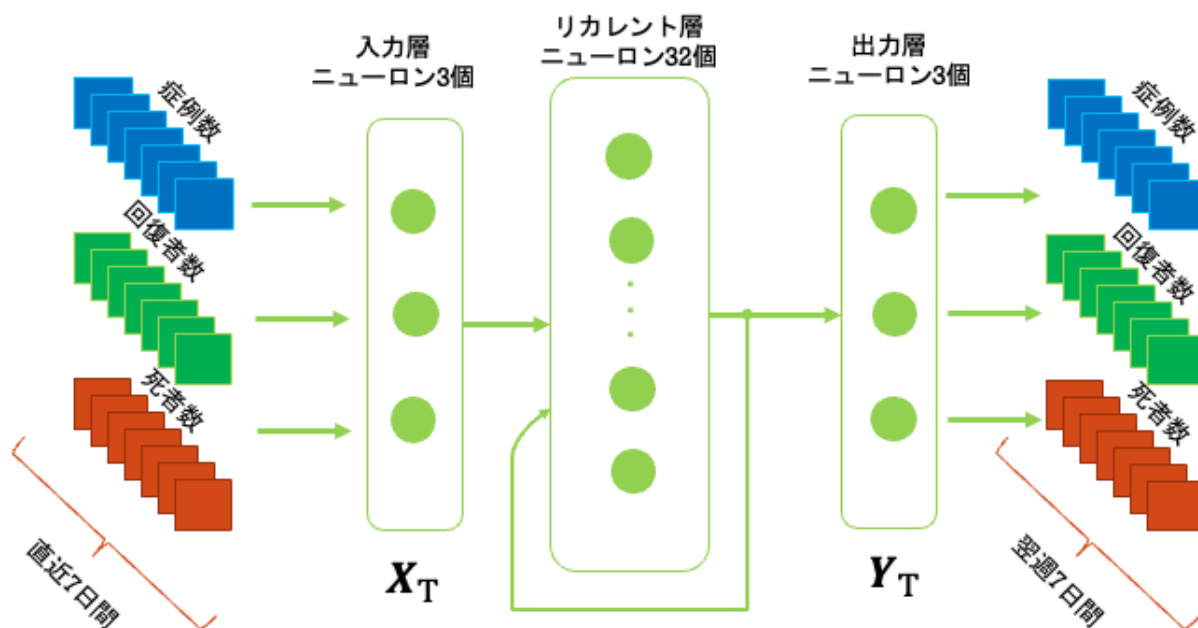


図 3.3: 構築した3次元入力3次元出力のRNN

### 3.3. 実験2における実験条件

#### 3.3.1. データ収集

実験2で使用した第7波までのCOVID-19に関するデータセットは、日本国の厚生労働省が提供するオープンデータセット [43] を元に編集・使用した。実験1で使用していたJHU提供のデータセットが各国のデータを毎日取得しているため途中でフォーマットが崩れPythonでのデータ読み込みが難しくなってしまった。そのため、実験1とデータセットの出典元が異なるのである。

図3.5に使用した厚生労働省のオープンデータセットを示す。

厚生労働省は、新規陽性者数の推移（日別）、入院治療等を要する者等推移、死亡者数（累積）等のデータを提供している。実験2で使用した厚生労働省が公開したデータセットの回復者数及び死亡者数のデータは累積データの為  $y_t - y_{t-1}$  のように前日との差分を取ることで新たに1日における新規回復者数、新規死者数のデータセットを作成し使用した。訓練データは、2020年5月10日～2022年10月16日の期間のデータを用い、テストデータは、2022年10月17日～同年10月23日の1週間のデータを使用した。なお、再帰型のニューラルネットワークの学習においては3.3.3節で述べるデータセットの作成時に7日間の対応関係が作成できなかった時系列後半のデータを使用せず、さらにそこから後半の1割を検証データとして使用した。

#### 3.3.2. Deep Learning 用入力データのスケールリング

再帰型ニューラルネットワーク用のデータセットには式3.1を用いてデータのスケールリングを行った。実験1とは異なり各変数毎の最大値（症例数=261004、回復者=273312、死者=347）と最小値（症例数=22、回復者=0、死者=0）で正規化を行った。また実験1同様、

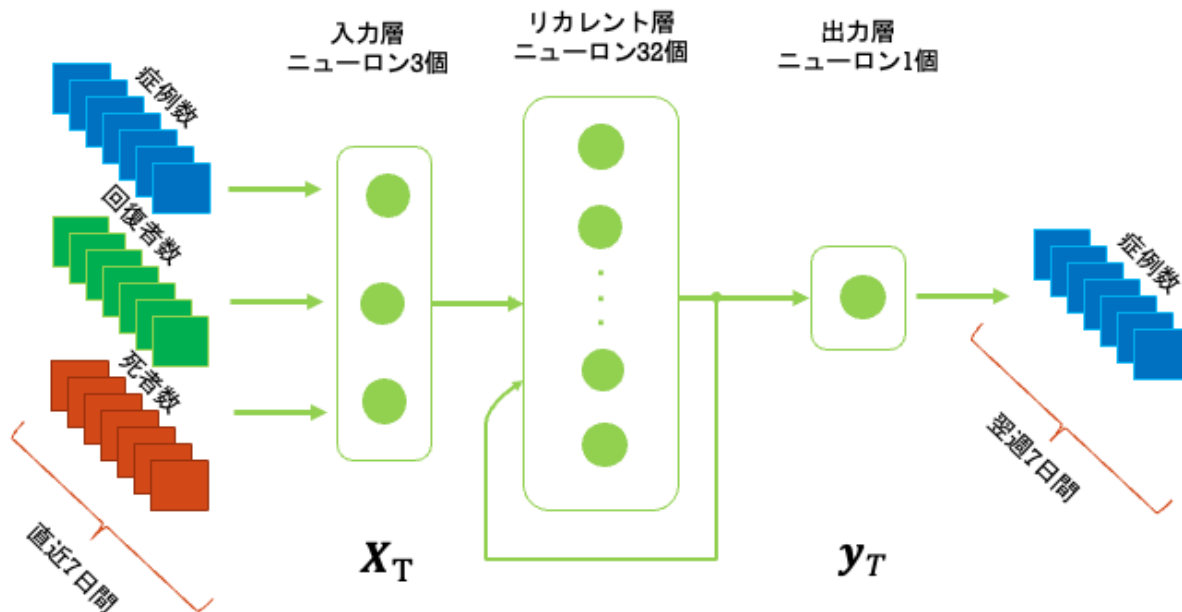


図 3.4: 構築した 3 次元入力 1 次元出力の RNN

RNN の中間層の活性化関数である  $\tanh$  関数の出力値に合わせ、正規化の範囲は  $-1 \sim +1$  に設定した。

### 3.3.3. データセットの作成

実験 2 では、Many To Many の入出力関係で再帰型のニューラルネットワーク (RNN、LSTM、Seq2Seq、Stacked LSTM) を構築した。Many To Many 用のデータセットは、図 2.6 のように 1 週間分の訓練データに対して次の 1 週間を教師データとして対応させる事で作成した。

### 3.3.4. モデル構築

統計モデルの ARIMA 及び SARIMA は実験 1 と同様に 3.1 節の表 3.1 に示した範囲で最小 AIC 推定を行い次数を決定しモデルを構築した。

実験 2 では機械学習モデルとして、RNN、LSTM、Seq2Seq、Stacked LSTM の 4 種類の 3 次元入力 3 次元出力の Many to Many の再帰型のニューラルネットワークを構築した。また、パラメータの詳細を表 3.3 に示す。全て同じ条件に設定し実験した。

### 3.3.5. 評価関数によるモデルの評価と比較

各モデルのテスト区間における予測評価を RMSE を用いて行なった。モデルからは、新規症例数、新規回復者数、新規死者数が出力されるが、予測の目的が症例数である為ここでは新規症例数のみの予測精度を算出した。実験 2 においても実験 1 と同様に Wu ら [9] の研究を参考にして式 3.2 を用いて ARIMA の RMSE からの性能差を算出した。



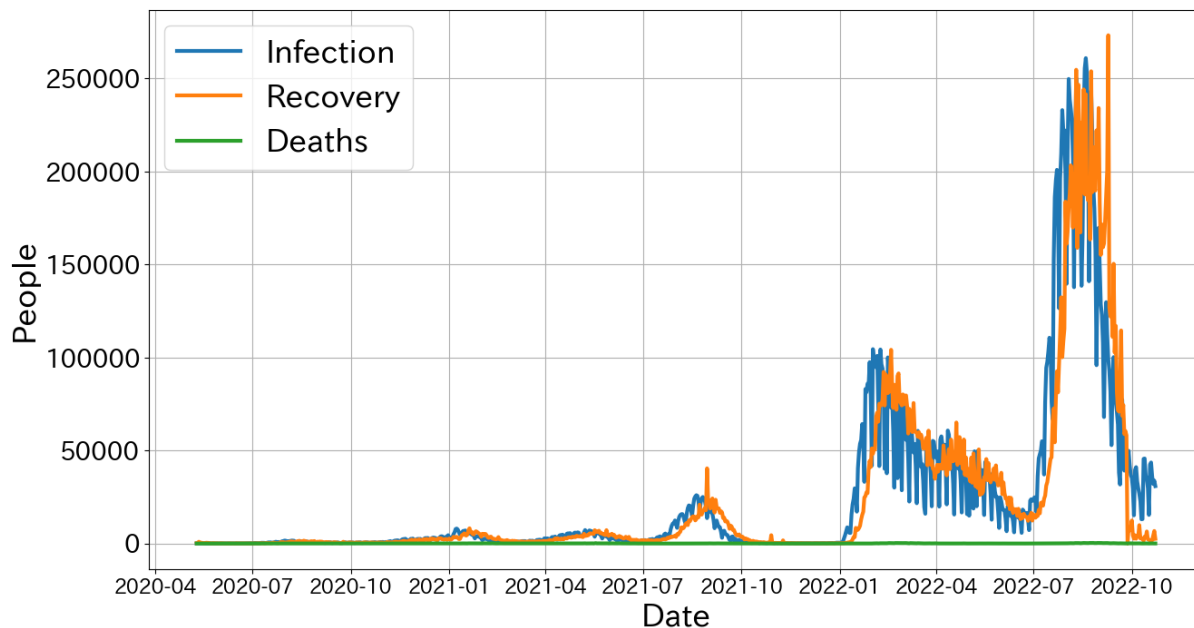


図 3.5: 厚生労働省提供の COVID-19 に関するオープンデータセット

表 3.3: 実験 2 における再帰型ニューラルネットワークの実験条件

モデル \ パラメータ	入力層 (個)
入力層のニューロン	3
中間層のニューロン	32
出力層のニューロン	3
最適化	Adam
エポック数	100
バッチサイズ	1

### 3.4. 実験 3 における実験条件

#### 3.4.1. データ収集

実験 3 で使用したサル痘データセットは、データサイエンスと機械学習のコンペティションを開催する Kaggle 上で公開されているデータセット [43] を元に編集・使用した。図 3.6 に使用した Kaggle 上で提供されているデータセットを示す。

Kaggle 上で提供されているデータセットには各国の新規感染者数が日別や週別に記録されている。本研究では、日別データが取得可能かつ感染者数の多い国を 3 カ国選び使用した。この 3 カ国の症例数データを選んだ理由は、Kaggle 上で提供されているデータセットにはまだサル痘に関するデータセットが少ないため回復者数や死者数のデータセットが存在しない為である。そこで、実験 2 で使用した同じアルゴリズム (RNN、LSTM、Seq2Seq、Stacked LSTM) を用いる事ができ、かつ、同じ入出力関係が構築できるように 3 カ国 (アメリカ、ドイツ、カナダ) の新規症例数を使用した。

訓練データは、2022 年 5 月 23 日～同年 8 月 19 日の期間のデータを用い、テストデータは、2022 年 8 月 20 日～同年 8 月 26 日の 1 週間のデータを用いた。なお、再帰型の



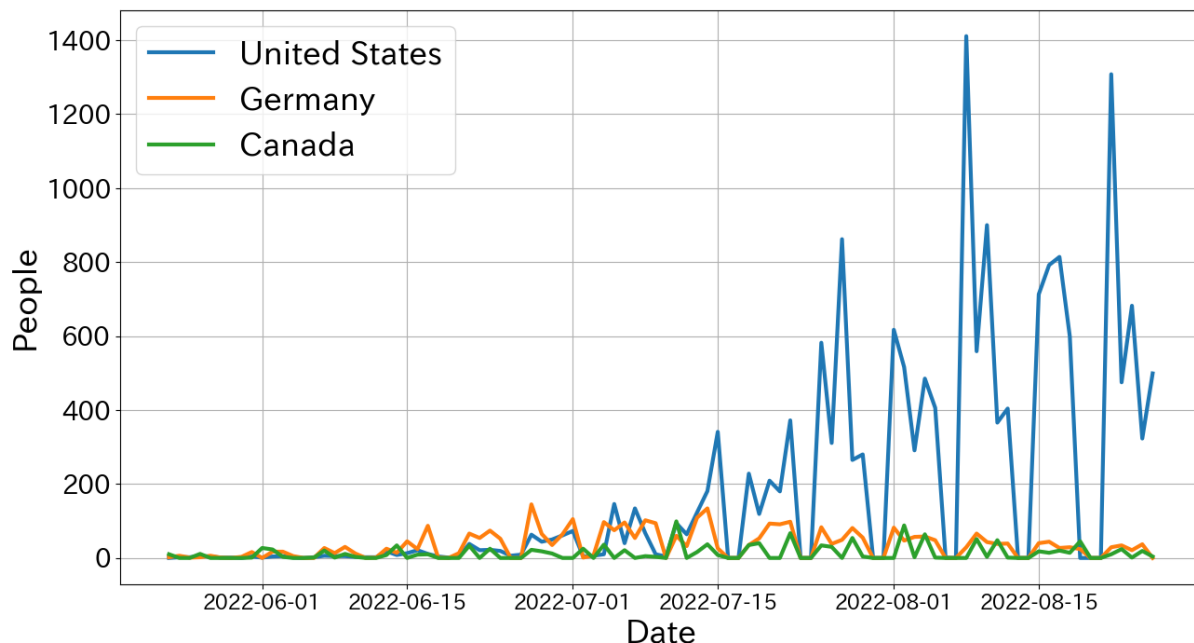


図 3.6: Kaggle 提供のサル痘のデータセット

ニューラルネットワークの学習においては 3.3.3 節で述べるデータセットの作成時に 7 日間の対応関係が作成できなかった時系列後半のデータを使用せず、さらにそこから後半の 1 割を検証データとして使用した。

### 3.4.2. Deep Learning 用入力データのスケーリング

実験 2 と同じく、再帰型のニューラルネットワーク用のデータセットには式 3.1 を用いてデータのスケーリングを行った。また実験 2 と同じく、各変数毎の最大値 (アメリカ = 1411 ドイツ = 145 カナダ = 99) と最小値 (感染者 = 0、回復者 = 0、死者 = 0) で正規化をおこなった。また実験 1 と 2 と同じく、RNN の中間層の活性化関数である tanh 関数の出力値に合わせ、正規化の範囲は  $-1 \sim +1$  に設定した。

### 3.4.3. データセットの作成

実験 3 では、実験 2 と同じく Many To Many を用いてモデルを構築した。Many To Many 用のデータセットは、図 2.6 のように 1 週間分の訓練データに対して次の 1 週間を教師データを対応させることで作成した。

### 3.4.4. モデル構築

統計モデルの ARIMA 及び SARIMA は、実験 1 及び実験 2 と同じく 3.1 節の表 3.1 に示した範囲で最小 AIC 推定を行い次数を決定しモデルを構築した。その結果は、4.4 節で述べる。

実験3の機械学習モデルとして実験2と同じく、RNN、LSTM、Seq2Seq、Stacked LSTMの4種類の3次元入力3次元出力の Many to Many の再帰型のニューラルネットワークを構築した。これら、4種類の再帰型のニューラルネットワークのパラメータの詳細は、全て表 3.3 と同じである。

### 3.4.5. 評価関数によるモデルの評価と比較

各モデルのテスト区間における予測評価を RMSE を用いて行なった。モデルからは3カ国全ての新規症例数が出力されるが、アメリカの新規症例数の予測が目的である為ここではアメリカの新規症例数のみの予測精度を算出した。実験2においても実験1及び実験2と同様に Wu ら [9] の研究を参考にして式 3.2 を用いて ARIMA の RMSE からの性能差を算出した。

## 第4章 結果と考察

### 4.1. COVID-19における統計モデルの探索結果

本研究において統計モデル (ARIMA、SARIMA) は、AICによる総当たり探索によって次数が決定された。第2波までの区間と第7波までの区間のCOVID-19新規症例数を、学習データセットとしてそれぞれARIMAとSARIMAに与えた。各データセットにおいて、学習データからモデルを推定し、AICを算出した。その結果、ARIMA及びSARIMAのAICが低いモデル5つを次の表4.1及び表4.2に与える。表4.1aは第2波までのデータでARIMAの次数探索を行った結果を、表4.1bは第7波までのデータでARIMAの次数探索を行った結果を示している。表4.2aは第2波までのデータでSARIMAの次数探索を行った結果を、表4.2bは第7波までのデータでSARIMAの次数探索を行った結果を示している。

表4.1より、第2波までのデータ及び第7波までのデータをARIMAで学習させた結果、次数の $(p=2, d=2, q=0)$ 及び $(p=3, d=2, q=1)$ が共通してAICが低かった。そこで、よりAICの低い $(p=2, d=2, q=0)$ をARIMAの次数として選択し、モデルを構築した。

また、表4.2より、第2波までのデータを学習させた結果及び第7波までのデータをSARIMAで学習させた結果、次数の $(p=1, d=0, q=1)(P=1, D=1, Q=0)$ 、 $(p=1, d=1, q=1)(P=1, D=1, Q=0)$ 及び $(p=1, d=2, q=1)(P=1, D=1, Q=0)$ が共通してAICが低かった。これらの次数では、季節項に関する次数 $(P=1, D=1, Q=0)$ が全て共通していた。そこで、2階差を取っている $(p=1, d=2, q=1)(P=1, D=1, Q=0)$ が定常過程になっている可能性が3つのモデルの中で最も高いと考え $(p=1, d=2, q=1)(P=1, D=1, Q=0)$ をSARIMAの次数として選択し、モデルを構築した。

表 4.1: ARIMA( $p, d, q$ ) の次数探索結果

(a) COVID-19の第2波までの区間において

次数	AIC
(3,2,0)	17.8
(2,2,0)	19.2
(3,2,1)	19.9
(2,2,2)	20.1
(2,2,1)	20.2

(b) COVID-19の第7波までの区間

次数	AIC
(1,2,0)	51.8
(2,2,0)	53.0
(3,2,1)	55.7
(1,2,2)	55.9
(1,2,3)	57.0

表 4.2: SARIMA( $p, d, q$ )( $P, D, Q$ )<sub>7</sub> のパラメータ探索結果

(a) COVID-19 の第 2 波までの区間において

次数	季節次数	AIC
(1,0,1)	(1,1,0)	8.0
(1,1,1)	(1,1,0)	8.0
(1,2,1)	(1,1,0)	8.0
(1,1,2)	(1,1,0)	10.0
(1,0,1)	(1,1,1)	10.0

(b) COVID-19 の第 7 波までの区間

次数	季節次数	AIC
(1,0,1)	(1,1,0)	8.0
(1,1,1)	(1,1,0)	8.0
(1,2,1)	(1,1,0)	8.0
(1,1,1)	(2,1,0)	10.0
(1,2,1)	(1,1,1)	10.0

## 4.2. 実験 1 : COVID-19 の第 2 波までを訓練させたモデルによる予測結果と考察

### 4.2.1. 結果

図 4.1 から図 4.5 に、COVID-19 の第 2 波までのデータを用いて訓練させた統計モデル (ARIMA、SARIMA) 及び 3 種類の入出力の RNN に訓練に用いたデータと同じ区間を予測させた結果を与える。図 4.1 から図 4.2 と図 4.3 から図 4.5 より、統計モデルの方が訓練データへのモデルのフィッティングが良いことが分かる。一方、3 種類の入出力の RNN モデルの予測結果には、第 1 波付近にバイアスが存在し、実際の観測値よりも予測値の方が大きな値を示していた。さらに、第 1 波周辺に観測値と予測値との間にラグが存在していた。

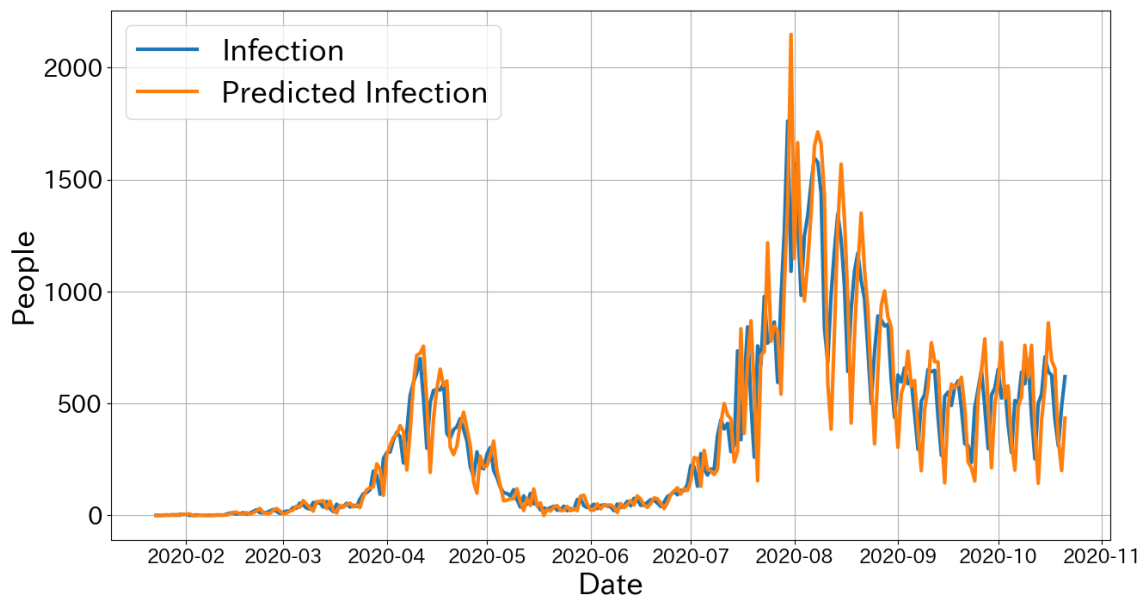


図 4.1: 第 2 波までの COVID-19 のデータを訓練させた ARIMA による訓練区間の予測結果

次に、統計モデル (ARIMA、SARIMA) 及び 3 種類の入出力の RNN で、モデル構築に用いなかったテスト区間のデータを予測を行った。そして、予測値と実際の観測値との誤差を RMSE で評価した結果を表 4.3 に与える。なお、表 4.3 中の 1to1、3to1、3to3 は RNN

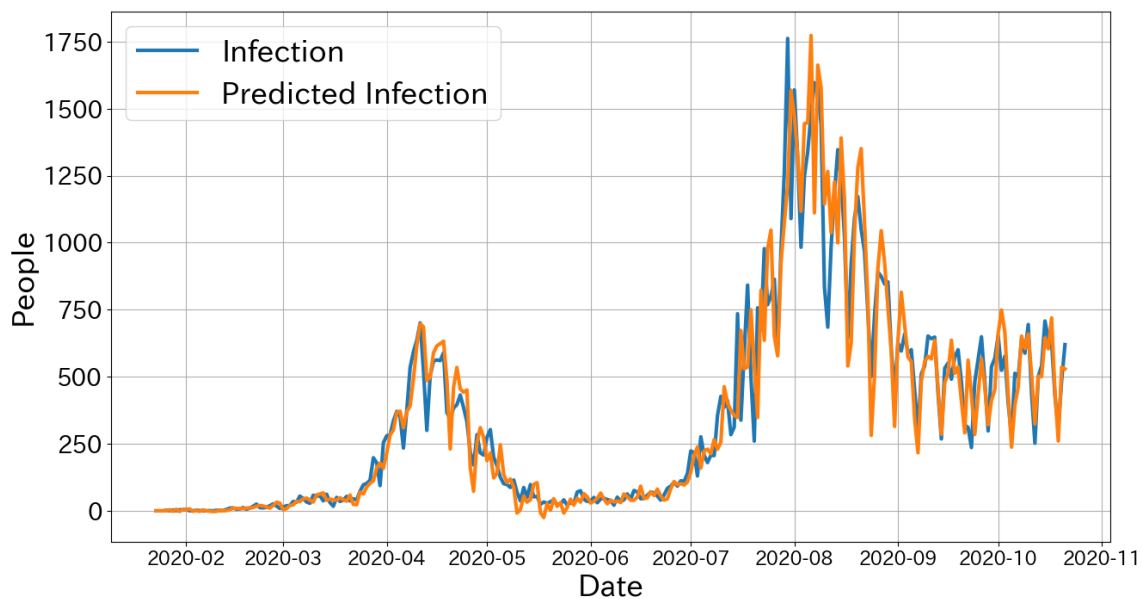


図 4.2: 第 2 波までの COVID-19 のデータを  
訓練させた SARIMA による訓練区間の予測結果

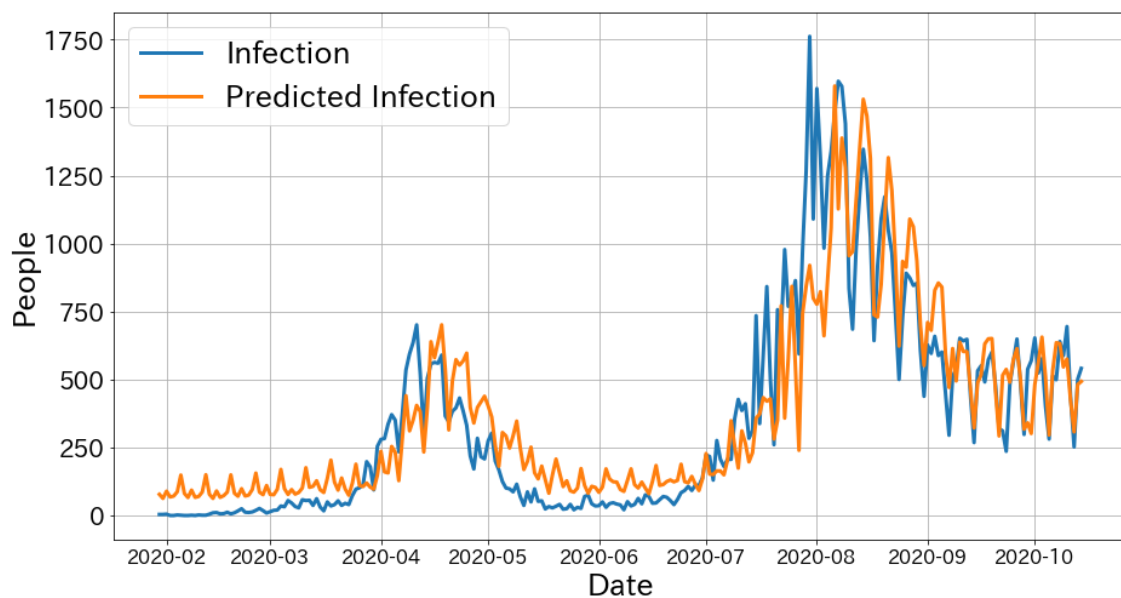


図 4.3: 第 2 波までの COVID-19 のデータを訓練させた  
1 次元入力 1 次元出力の RNN による訓練区間の予測結果

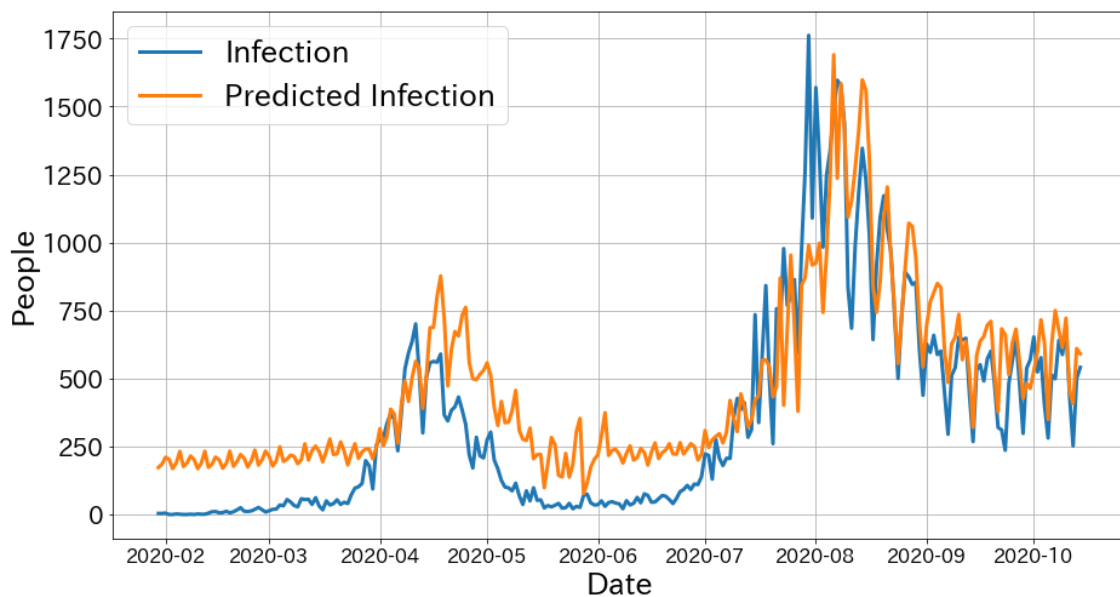


図 4.4: 第 2 波までの COVID-19 のデータを訓練させた  
3次元入力 1次元出力の RNN による訓練区間の予測結果

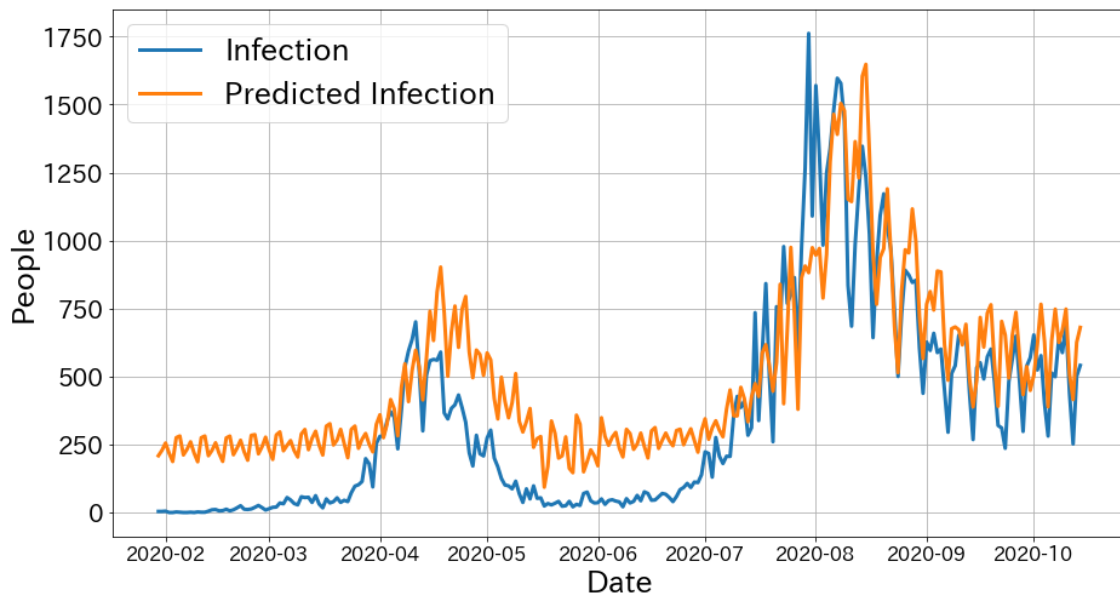


図 4.5: 第 2 波までの COVID-19 のデータを訓練させた  
3次元入力 3次元出力の RNN による訓練区間の予測結果

の入出力関係を指し、それぞれ 1 次元入力 1 次元出力の RNN、3 次元入力 1 次元出力の RNN、3 次元入力 3 次元出力の RNN を意味する。また、表 4.3 の差 (%) は、ARIMA の RMSE を基準として何%誤差が増減したかを意味する。なお、差 (%) は式 3.2 によって算出された。

表 4.3: COVID-19 の第 2 波までのデータを訓練させたモデルのテスト区間の症例数予測結果の評価

	ARIMA	SARIMA	1to1	3to1	3to3
RMSE(人)	464	67	110	59	47
差 (%)	0.0	-85.6	-76.3	-87.3	-89.9

表 4.3 より、統計モデル (ARIMA、SARIMA) の中では SARIMA の方がテスト誤差を最小化できていた。RNN の中では、1 次元入力 1 次元出力 > 3 次元入力 1 次元出力 > 3 次元入力 3 次元出力の順に RMSE が小さかった。全モデルの中で、感染者数の予測誤差を最小化できたのは、3 次元空間から 3 次元空間に時系列データを写像した RNN であった。また表 4.3 中の差 (%) より、統計モデルの ARIMA の RMSE との相対的な予測誤差 (RMSE) の増減率においても、最も減少したモデルは 3 次元空間から 3 次元空間に時系列データを写像した RNN であった。RNN において、3 次元入力 1 次元出力のモデルは 3 次元入力 3 次元出力のモデルよりも精度が高くなることを期待していた。しかし、実際には 3 次元入力 3 次元出力の RMSE が小さかった。ただし、どちらのモデルとも統計モデルの ARIMA、SARIMA と比べた際は RMSE を小さくできていた。入出力にベクトルを用いたこれら 2 つの RNN は、どちらも ARIMA から約 90% 誤差を減少していた。1 次元入力 1 次元出力の RNN は、ARIMA よりも 85.5% RMSE を減少できていた。しかし、その減少量は SARIMA よりも小さかった。

ここで、表 4.4 に Wu[9] らの研究で示された ARIMA の RMSE に対するインフルエンザ様疾患発生予測モデルの相対的な減少率を示す。

表 4.4: Wu[9] らのインフルエンザ様疾患発生予測モデルの予測評価

	ARIMA	LSTM	SeqToSeq+Attention	Transformer
RMSE(%)	1.020	0.807	0.642	0.588
Gap(%)	0.0	-20.9	-37.1	-42.4

次に、表 4.3 に与えた本研究の実験 1 の結果と表 4.4 の Wu らの実験結果を比較した。Wu らのモデルはインフルエンザ様疾患の前週の発生数との比率のデータで構築されている。そのため、単純比較はできない。しかし、今回作成したモデルの中で最も予測誤差の小さい 3 次元入力 3 次元出力の RNN は、Wu らの構築した Transformer よりも約 47.5% 高い精度を示した。

#### 4.2.2. 考察

図 4.4 から図 4.5 より、入力を多変数とした RNN はどれも訓練させた期間を予測した場合において予測精度が悪かった。これは、スケーリングの最大値と最小値の設定に問題

があったと考える。実験1では、全てのデータの最大値である回復者数の3,941人と全てのデータの最小値である0人で正規化を行なった。この時、使用したCOVID-19の2波までのデータは図3.1に示した通り、回復者の波と感染者数の波にラグが存在する。そのため、正規化後のデータを学習した際にトレンドが回復者数のトレンドに近い値を学習した。その結果、第1波周辺に存在するような観測値よりも予測値の方が大きい現象や、予測値のピークが観測値のピークの後に遅れて予測される現象が起きたと考える。そこで、今後は安定した予測の為に、各変数の最大値及び最小値を用いて正規化を行う必要があると考える。

次に、表4.3の結果について考察する。まず、統計モデル(ARIMA、SARIMA)において、SAIRMAがARIMAに対して約85.5%RMSEを減少した。その要因は、SARIMAが季節性をモデルに取り込めたからだと考える。図4.6に、テスト区間をARIMA及びSARIMAで予測した結果を与える。

図4.6aより、ARIMAによる予測は右肩上がりの予測になっている。これは、テストデータの区間が第3波に向けた上昇トレンドの区間であることから、ARIMAがトレンド成分を学習した為だと考える。一方で、図4.6bのSARIMAによる予測の結果は、1週間周期の振幅を追従できている。

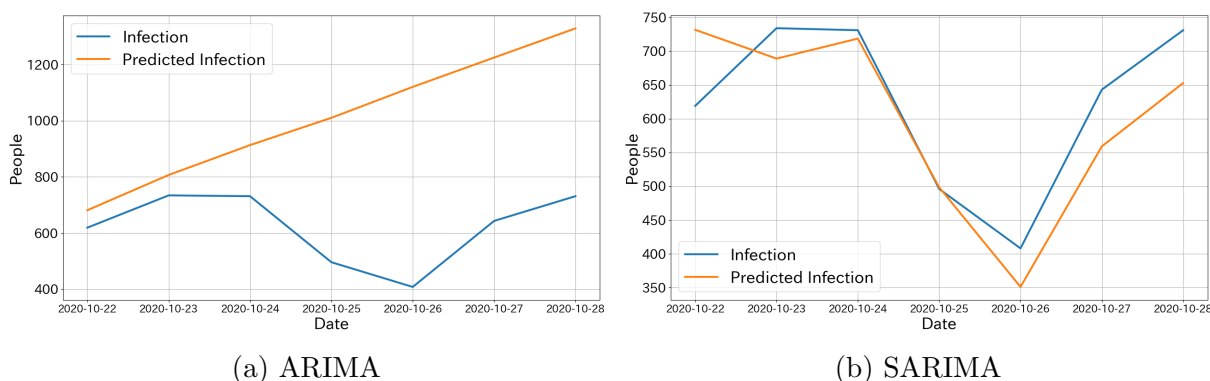


図 4.6: テスト区間を第2波までのCOVID-19のデータを訓練させた統計モデルによって予測した結果

ここで図4.7に、訓練を行った区間をARIMAモデルとSARIMAモデルによって予測した予測値と観測値の残差の自己相関関数のコレログラムを与える。コレログラムは、各ラグにおける自己相関をプロットした図のことを言う。図4.7に示した青色の区間は、95%の信頼区間でこの区間を超えた自己相関は統計的に有意な自己相関とされる。

図4.7aより、ARIMAの残差には7日間周期の季節成分が含まれていることがわかる。その為、モデルに季節的な周期成分を含むことができていないと考えられる。また、図4.7bの結果より、SARIMAの残差には統計的に有意な自己相関はあるもののほとんどその係数は小さい。その為、モデルに季節成分を取り込んでいると考えられる。このことから、ARIMAはトレンド成分のみを学習し、SARIMAはトレンドに加えて季節成分を学習できていると言える。

次に、3種類の入出力を行なったRNNモデルによってテスト区間を予測した結果について考察する。

最も精度の高いRNNは、3次元空間(入力)から3次元空間(出力)へ写像したモデルであった。3次元空間(入力)から1次元空間(出力)に写像したRNNと比較すると、



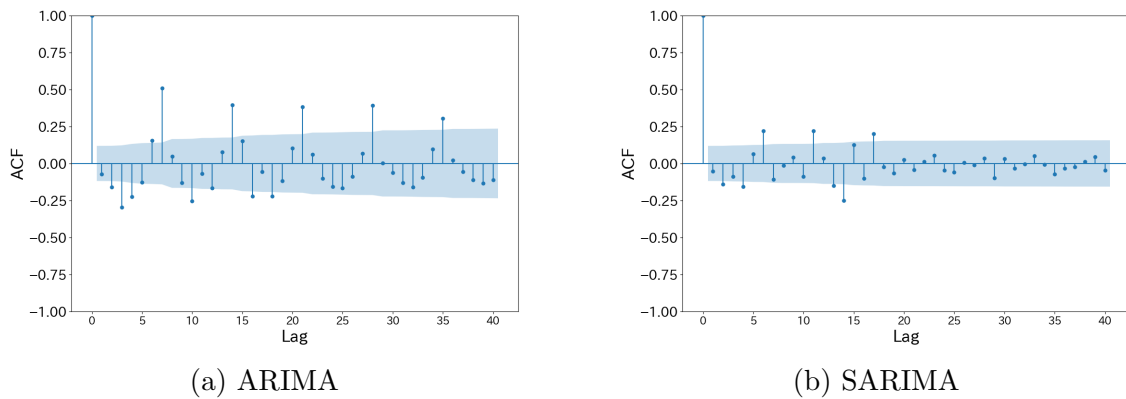


図 4.7: 第 2 波までの COVID-19 のデータを訓練させたモデルの残差における自己相関関数のコレログラム

高次元空間から低次元空間に写像するよりも同次元の空間に写像した方が高い予測精度を得られることがわかった。これは、RNN の中間層が式 2.20 のように定義される事に起因すると考える。RNN の中間層では、1 つ前の時刻の中間層の出力  $h_{t-1}$  が  $x_t$  と共に関数  $f$  の入力となる。従って、関数が  $f(x_t, h_{t-1})$  で与えられ、高次元空間（入力）から低次元空間（出力）に写像した場合、写像後に情報量の欠損が起きる。実験 1 の場合、感染者数、回復者数、死者数の 3 次元空間から感染者数の 1 次元空間に写像している。そのため、回復者数と死者数に関する情報が入力空間のまま保持されていない。従って、その情報を再起的に入力した場合に RNN がうまく回復者数や死者数の情報まで時系列として学習できなくなると考える。一方で、3 次元空間（入力）から 3 次元空間（出力）へ写像したモデルでは全ての変数の情報量がそのまま写像され、再帰入力されることで時系列情報の欠損が起きずに RNN によって全ての変数の時系列情報が学習されたと考える。

今回作成した 3 種類の RNN は、どのモデルも Wu ら [9] のインフルエンザ様疾患発生予測モデルである Tansformer の精度を大きく超える結果となった。この理由として、RNN の中間層における活性化関数である tanh 関数が Additive Attention[44] として機能したことが考えられる。ここでの、Additive Attention とは、関数などのニューラルネットワーク内部の機構で特徴量を強調する機能である。今回の場合、この Additive Attention が自己の系列の特徴量を強調する Self-Attention として機能したと考える。従って以降では、Additive Attention のことを Self-Attention と記述する。

ここで、図 4.8 に tanh 関数の入出力を示す。図 4.8 のように、tanh 関数は入力が  $-2$  から  $+2$  の際に最も出力値が変動する。実験 1 で扱った JHU 提供のデータセットは、式 3.1 によって値が  $-1$  から  $+1$  に正規化されている。そのため、入力データが少しでも大きな値、もしくは小さな値になった時に出力値が  $-1$  もしくは  $+1$  に近づく。従って、正規化した入力データは tanh によって特徴量が強調されたと考える。

実験 1 の場合では、 $-1$  から  $+1$  に正規化した入力データが中間層の tanh 関数の Self-Attention によって重要な相関関係を持つ日を 1 日単位で次のニューロンに流すことによって良いモデルが構築されたと考える。

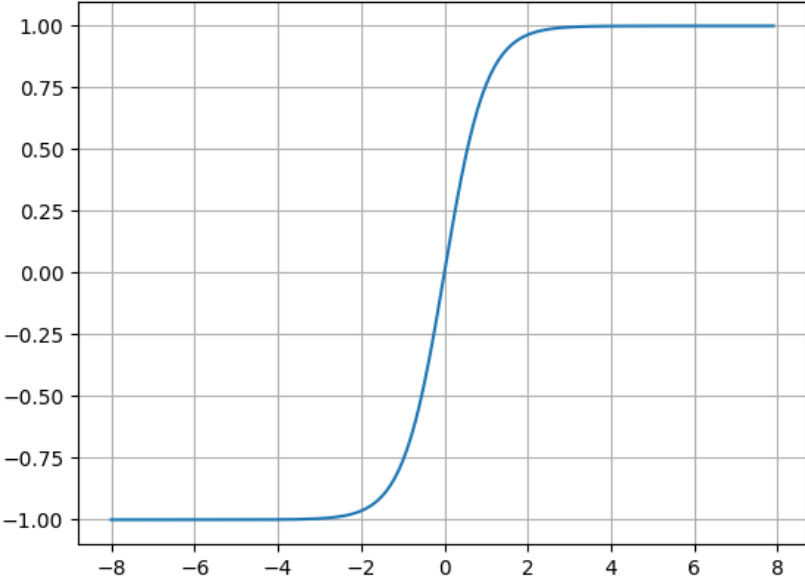


図 4.8: Tanh 関数の入出力

### 4.3. 実験2：COVID-19の第7波までを訓練させたモデルによる予測結果と考察

#### 4.3.1. 結果

図4.9から図4.14に、第7波までのデータを用いて学習させた統計モデル（ARIMA、SARIMA）及び4種類の再帰型のニューラルネットワーク（RNN、LSTM、Stacked LSTM、Seq2Seq）モデルに訓練に訓練区間を予測させた結果を与える。図4.9から図4.10及び図4.11から図4.14より、統計モデルの方が訓練データへのモデルのフィッティングが良いことが分かる。一方、4種類の再帰型のニューラルネットワークモデルの予測結果には、第5波より前の波形にバイアスが存在し、実際の観測値よりも予測値の方が大きな値を示していた。また、図4.13と図4.14に示した層数の多いStacked LSTMとSeq2Seqでは、観測値と比べた際に第7波の予測値が大きく下回っていた。

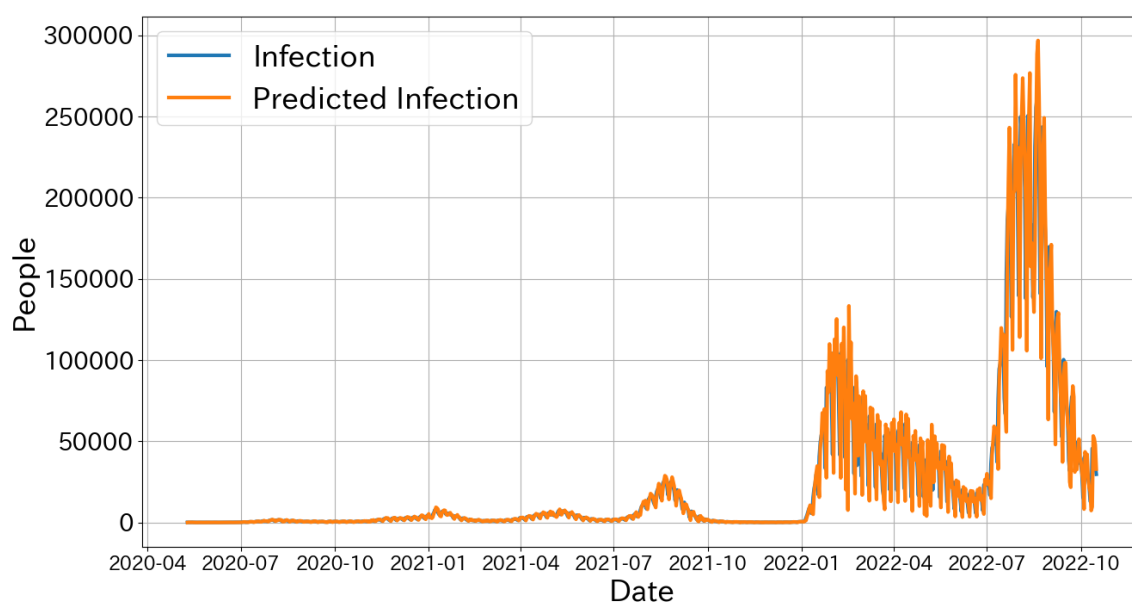


図 4.9: 第7波までの COVID-19 を訓練させた ARIMA による訓練区間の予測結果

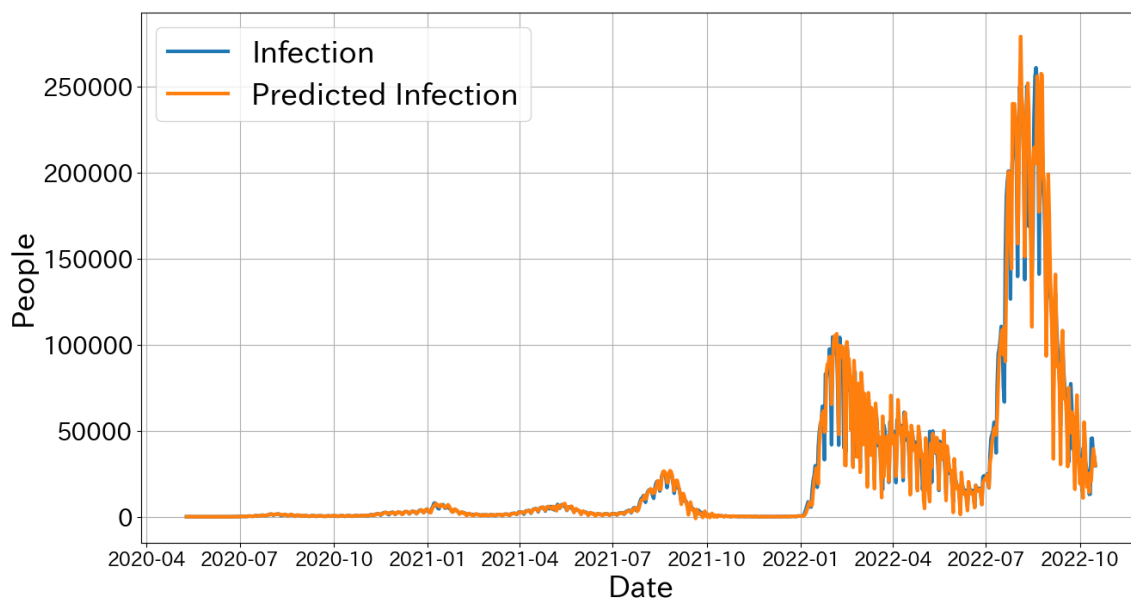


図 4.10: 第 7 波までの COVID-19 を訓練させた SARIMA による訓練区間の予測結果

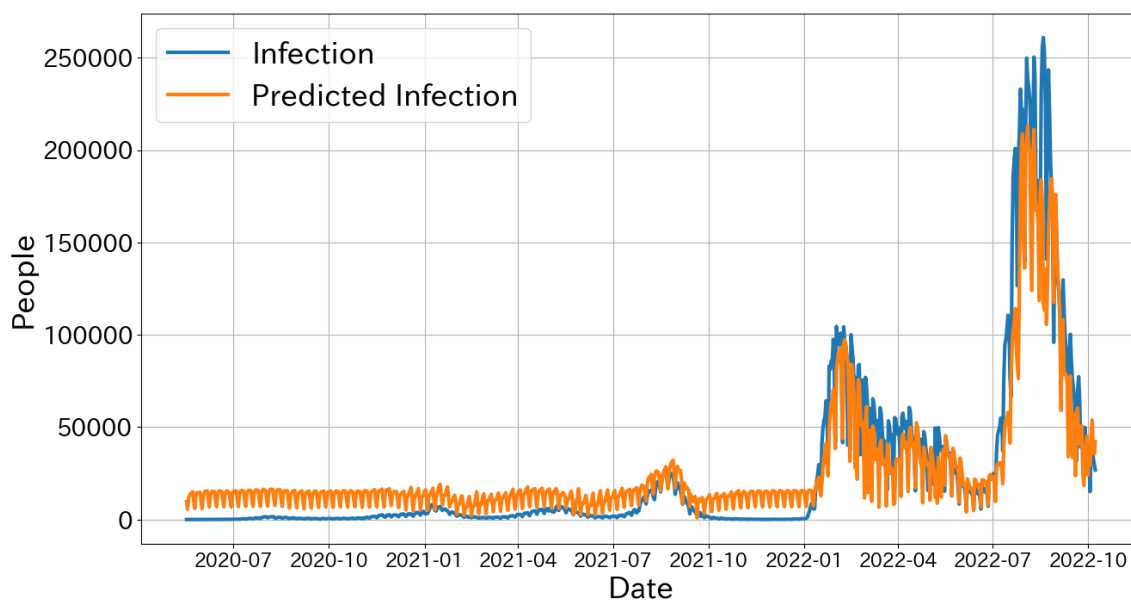


図 4.11: 第 7 波までの COVID-19 のデータを訓練させた RNN による訓練区間の予測結果

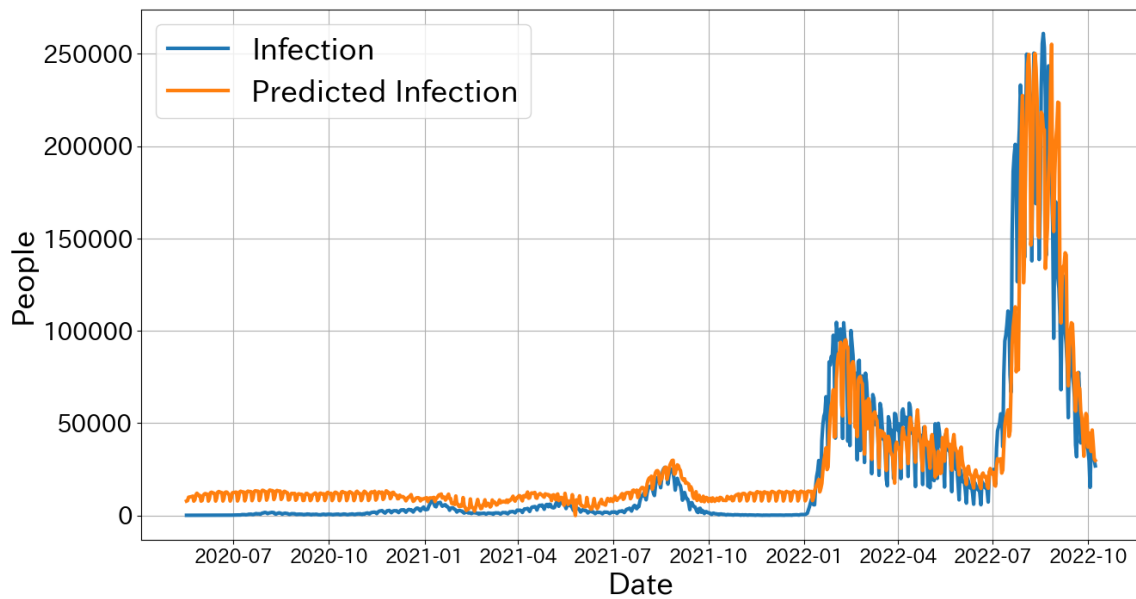


図 4.12: 第 7 波までの COVID-19 のデータを訓練させた LSTM による訓練区間の予測結果

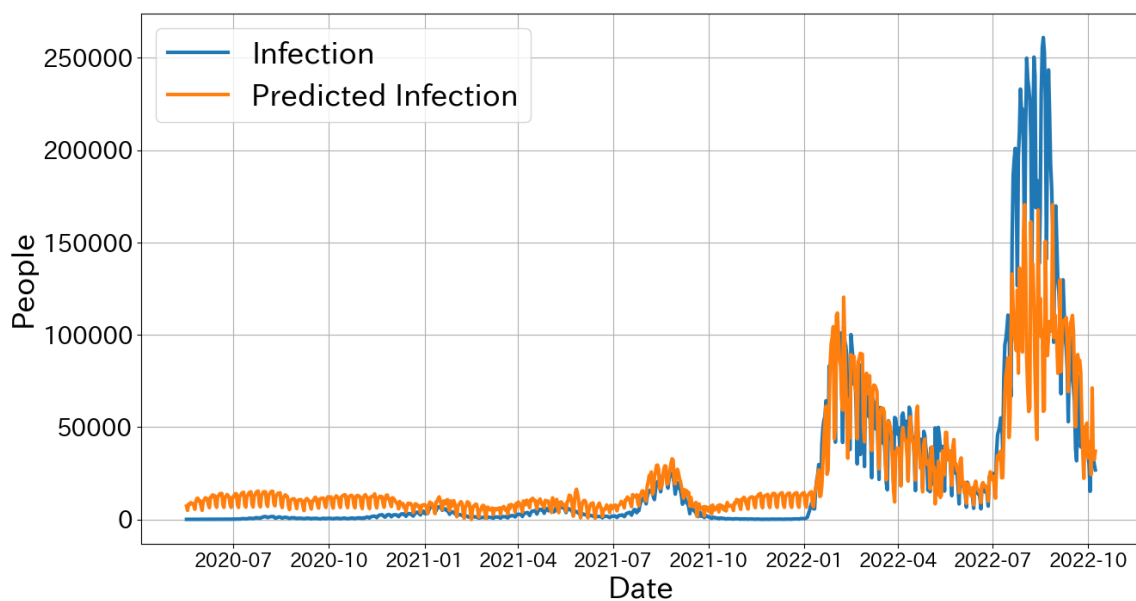


図 4.13: 第 7 波までの COVID-19 のデータを訓練させた Stacked LSTM による訓練区間の予測結果

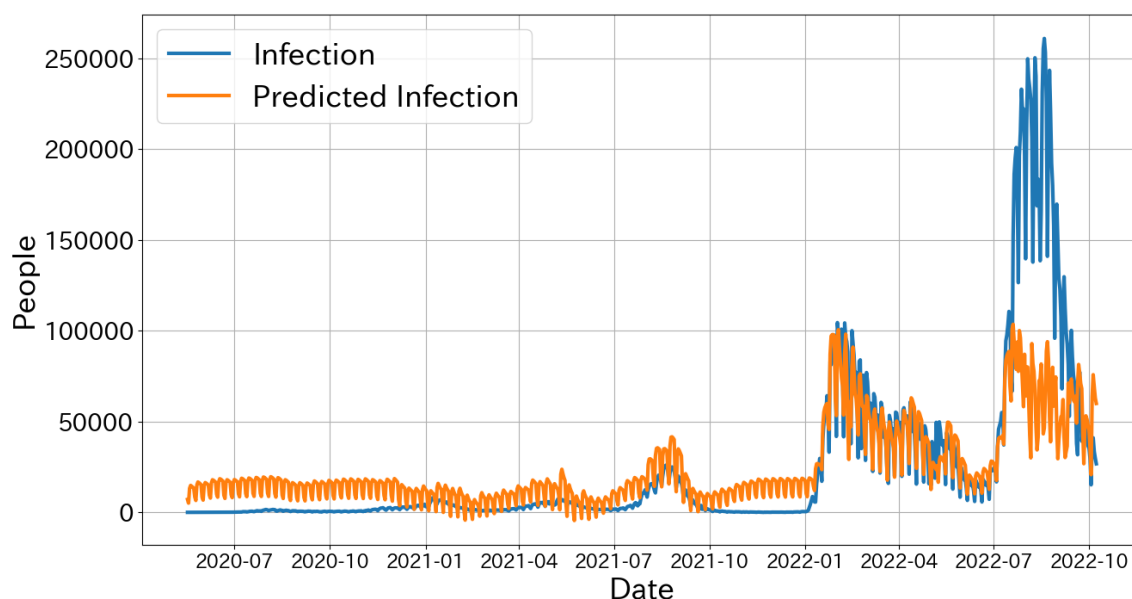


図 4.14: 第 7 波までの COVID-19 のデータを訓練させた Seq2Seq による訓練区間の予測結果

次に、統計モデル (ARIMA、SARIMA) 及び 4 種類の再帰型のニューラルネットワークで、モデル構築に用いなかったテスト区間の予測を行った。そして、予測値と実際の観測値との誤差を RMSE で評価した結果を表 4.5 に与える。なお、表 4.5 中の Stacked は中間層にリカレント層を 2 層用いた Stacked LSTM を意味する。また、表 4.5 の差 (%) は ARIMA の RMSE を基準として何%誤差が増減したかを意味する。差 (%) は、式 3.2 によって算出された。

表 4.5: 第 7 波までの COVID-19 のデータを訓練させたモデルのテスト区間の症例数予測結果の評価

	ARIMA	SARIMA	RNN	LSTM	Seq2Seq	Stacked
RMSE(人)	28167	6707	14569	10827	25776	17419
差 (%)	0.0	-76.2	-48.3	-61.6	-8.5	-38.2

表 4.5 より、統計モデル (ARIMA、SARIMA) の中では SARIMA の方がテスト誤差を最小化できていた。4 種類の再帰型のニューラルネットワークモデルの中で、最も小さな予測誤差を示したモデルは単層の LSTM であった。全モデルの中で感染者数の予測誤差を最小化できたモデルは、SARIMA であった。また表 4.5 中の差 (%) より、統計モデルの ARIMA の RMSE との相対的な予測誤差 (RMSE) の増減率においても、最も減少したモデルは SARIMA であった。

再帰型のニューラルネットワークにおいて、Seq2Seq や Stacked LSTM などでは層数が増えることで単層の RNN や LSTM よりも表現力が高くなり長期的トレンドをモデルに含む事ができ予測精度が高くなることを期待していた。しかし、Seq2Seq と Stacked LSTM の層数が多いモデルよりも層数が少ないモデル (単層の RNN と LSTM) の方が RMSE を小さくできていた。Seq2Seq を除いた 3 種類の再帰型のニューラルネットワークモデルはどれも ARIMA からの約 40% 以上誤差を減少していた。

次に、表 4.5 に与えた本研究の実験 2 の結果と表 4.4 の Wu らの実験結果を比較した。Wu らのモデルは、インフルエンザ様疾患の前週の発生数との比率のデータで構築されている。その為、単純比較はできない。しかし、今回作成したモデルの中で最も予測誤差の小さい SARIMA モデルでは、Wu らの構築した Transformer よりも 33.8%高い精度を示した。また、4 種類の再帰型のニューラルネットワークの中で最も予測誤差の小さい LSTM では、Wu らの構築した Transformer よりも 19.2%高い精度を示した。

### 4.3.2. 考察

図 4.11 から図 4.14 より、4 種類の再帰型のニューラルネットワークモデル (RNN、LSTM、Seq2Seq、Stacked LSTM) は、全て第 5 波までの波形に対してロバストなモデルとなり観測値よりも大きな値を予測している。これはリカレント層の活性化関数である tanh 関数の影響だと考える。実験 2 は、入力データを  $-1$  から  $+1$  で正規化をしている。そのため、感染者数が少ないとき (正規化前の値が 0 付近のデータ)、モデルの学習時に各アルゴリズム入力される値は  $-1$  に近い値になる。図 4.8 を見ると、 $-1$  に近い値が入力された場合出力である y 軸の値はほとんど変動せず  $-0.75$  に近い値を出力する。この現象は、図 4.11 に示した RNN で顕著に観測された。これに比べて、図 4.12 から図 4.14 に示した LSTM を基本としたモデルでは、バイアスが出現した区間における振幅は小さかった。これは、LSTM が式 2.24 と式 2.26 にあるように入力情報を sigmoid 関数によって 0 から 1 の値に変換し、長期記憶されたセルステート  $C_t$  の出力を調節している事が原因と考える。4.2 節で述べたように、 $-1$  から  $+1$  に正規化した入力データが中間層の tanh 関数の Self-Attention[44] によって 1 日単位で重要な相関関係を持つ日を次のニューロンに流すことが可能なのは入力値が 0 付近に存在する場合である。従って、今回の入力データのように正規化後の最大値 ( $+1$ ) や最小値 ( $-1$ ) に対してデータ量の偏りがある場合はロバストに反応するモデルが構築されると考える。

次に、表 4.5 の結果について考察する。まず統計モデルの ARIMA 及び SARIMA に関して考察する。SAIRMA が ARIMA に対して 76.2%RMSE を減少した。この要因は、季節性をモデルに取り込めた結果だと考える。以下図 4.15 に、テストデータを入力とした ARIMA 及び SARIMA の予測結果を与える。

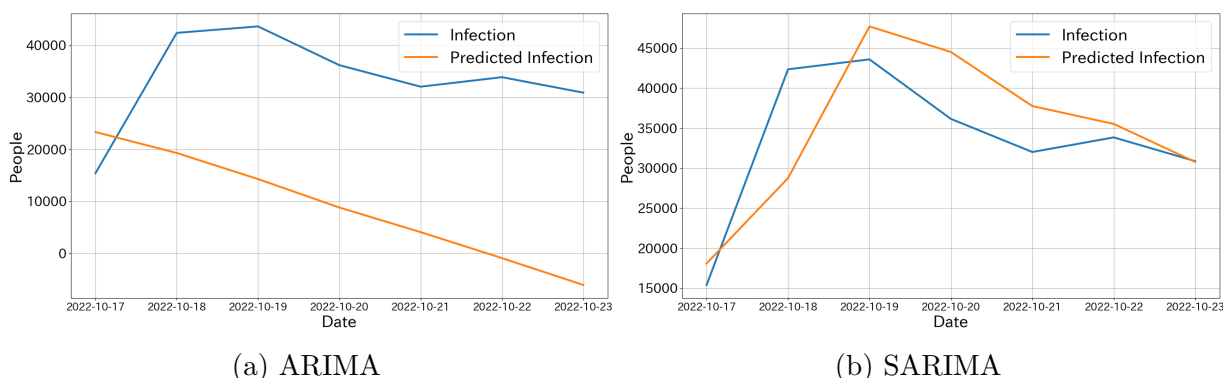


図 4.15: テスト区間を第 7 波までの COVID-19 のデータを訓練させた統計モデルによって予測した結果

図 4.15a より、ARIMA の予測は右肩下がりの予測になっている。これは、テストデー



夕の区間が第7波収束に向けた下降トレンドの区間であることから、ARIMAがこの下降トレンドを学習した事に起因すると考える。一方で、図4.15bのSARIMAの予測結果は1週間周期の振幅を追従できている。

ここで図4.16に、訓練に使用した区間をARIMAモデルとSARIMAによって予測した予測値と観測値の残差の自己相関関数のコレログラムを与える。コレログラムは、各ラグにおける自己相関をプロットした図のことを言う。図4.7に示した青色の区間は、95%の信頼区間でこの区間を超えた自己相関は統計的に有意な自己相関とされる。

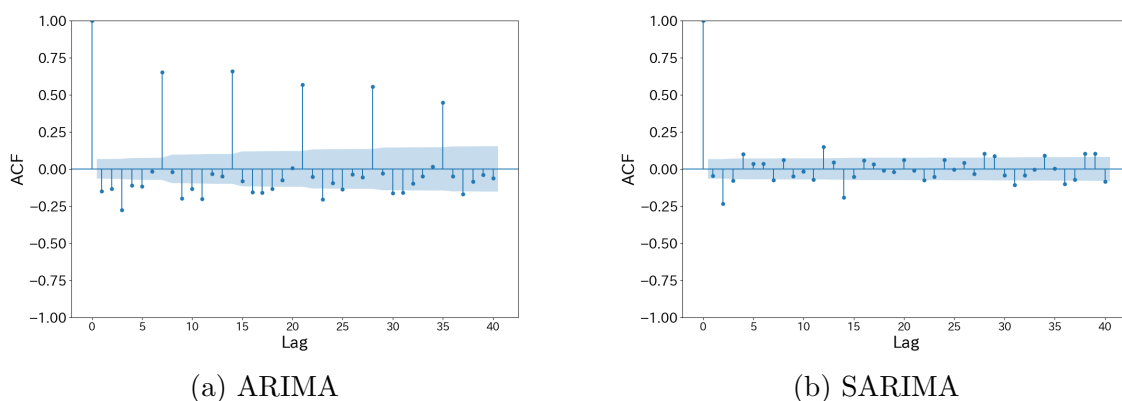


図4.16: 第7波までのCOVID-19のデータを訓練させたモデルの残差における自己相関関数のコレログラム

図4.16aより、95%の信頼区間から大きく出ている値が7日間周期に存在することがわかる。従って、ARIMAの残差には7日間周期の季節成分が残っており、モデルに季節的な周期成分を含むことができていないと考えられる。また、図4.16bの結果から、SARIMAの残差では、統計的に有意な自己相関がほとんど無く、モデルに季節成分を取り込んでいると考えられる。このことから、ARIMAはトレンド成分のみを学習し、SARIMAはトレンドに加えて季節成分を学習できていると言える。

次に、4種類の再帰型ニューラルネットワークモデル(RNN、LSTM、Seq2Seq、Stacked LSTM)にテストデータを入力し、予測した結果について考察する。

Stacked LSTMやSeq2Seqでは、単層のモデルよりも精度が高い事を期待していた。しかし、表4.5より単層のRNN及び単層のLSTMの方が高い精度を示した。この原因として、単層のRNNと単層のLSTMでは、中間層(リカレント層)における活性化関数であるtanh関数がSelf-Attentionとして機能したことが考えられる。このことより、tanh関数によるSelf-Attentionの効果は単層の時のみ機能すると考える。また、Seq2Seqにおいては図4.14の最後の1割の検証区間を見ると著しく予測精度が悪い。これは、検証区間にスケリング後の値が0以上のサンプルが偏っているため1週間における要約値であるコンテキストベクトル $C_t$ を用いて学習を行うことが難しかったと考える。

今回作成したSARIMA、単層のRNN、そして単層のLSTMは、Wuら[9]のインフルエンザ予測用Transformerモデルの精度を大きく超える結果となった。この理由として、SARIMAでは7日間の季節性が取り込めたことが考えられる。単層のRNNと単層のLSTMでは、中間層(リカレント層)における活性化関数であるtanh関数がSelf-Attentionとして機能したことが考えられる。単層のRNNと単層のLSTMにおいて、LSTMの方が予測誤差が小さかったのはLSTM Cellの機能的特徴が原因と考えられる。LSTMは、上記



で述べたとおり最終出力の算出時に保持している長期トレンドに対して入力値に sigmoid 関数を通した 0 から 1 の値を乗算する。その為、入力データの値に偏りが長期にわたって存在する場合 RNN よりも LSTM の方が精度の高いモデルが構築されたと考える。

## 4.4. サル痘のデータにおける統計モデルの探索結果

サル痘予測モデルにおいても、AIC を基準に総当たり探索を行うことで統計モデルの ARIMA 及び SARIMA の次数を決定した。

表 4.6 に、AIC を基準とした次数探索の結果 AIC が低かった 5 つの次数を与える。

表 4.6: サル痘のデータにおける統計モデルの次数探索結果

(a) ARIMA( $p, d, q$ )

次数	AIC
(3, 2, 0)	16.3
(3, 2, 2)	16.4
(3, 2, 3)	18.4
(1, 2, 0)	27.9
(2, 2, 0)	28.9

(b) SARIMA( $p, d, q$ )( $P, D, Q$ )<sub>7</sub>

次数	季節次数	AIC
(1, 0, 1)	(1, 1, 0)	8.0
(1, 1, 1)	(1, 1, 0)	8.0
(1, 2, 1)	(1, 1, 0)	8.0
(1, 2, 1)	(1, 1, 1)	10.0
(1, 0, 1)	(1, 1, 1)	10.0

表 4.6a より、サル痘のデータを ARIMA で学習させた結果では、次数が ( $p = 3, d = 2, q = 0$ ) のとき、AIC が最も低かった。従って、( $p = 3, d = 2, q = 0$ ) を ARIMA の次数として選択しモデルを構築した。

また、表 4.6b より、サル痘のデータを SARIMA 学習させた結果では、次数が ( $p = 1, d = 0, q = 1$ )( $P = 1, D = 1, Q = 0$ )、( $p = 1, d = 1, q = 1$ )( $P = 1, D = 1, Q = 0$ )<sub>7</sub> 及び ( $p = 1, d = 2, q = 1$ )( $P = 1, D = 1, Q = 0$ )<sub>7</sub> のとき、AIC が最も低かった。これらは、季節項に関する次数 ( $P = 1, D = 1, Q = 0$ ) が全て共通していた。そこで、3 つのモデルの中で 2 階差を取っている ( $p = 1, d = 2, q = 1$ )( $P = 1, D = 1, Q = 0$ )<sub>7</sub> の定常過程になっている可能性が最も高いと考え ( $p = 1, d = 2, q = 1$ )( $P = 1, D = 1, Q = 0$ )<sub>7</sub> を SARIMA の次数として選択しモデルを構築した。

## 4.5. 実験 3: サル痘のデータを訓練させたモデルの予測結果と考察

### 4.5.1. 結果

図 4.17 から図 4.22 に、サル痘ウイルスのデータを学習させた統計モデル (ARIMA、SARIMA) 及び 3 次元 (アメリカ、ドイツ、カナダの 3 国の症例数) 入力 3 次元 (アメリカ、ドイツ、カナダの 3 国の症例数) 出力の再帰型のニューラルネットワーク (RNN、LSTM、Seq 2 Seq、Stacked LSTM) に、訓練に用いた区間を予測させた結果を与える。

図 4.17 より、ARIMA モデルは初期の時系列にある感染者数が 1 桁代のデータに対してはフィッティングができていたが、振幅が大きくなる時系列の後半では短期的な負のトレンドを大きく見積もってしまい感染者数が負の値になった。さらに、予測した値と観測値には数日のラグが存在した。一方で、図 4.18 から SARIMA は時系列後半においても過剰に短期的な負のトレンドに対してロバストに反応する事はなかった。

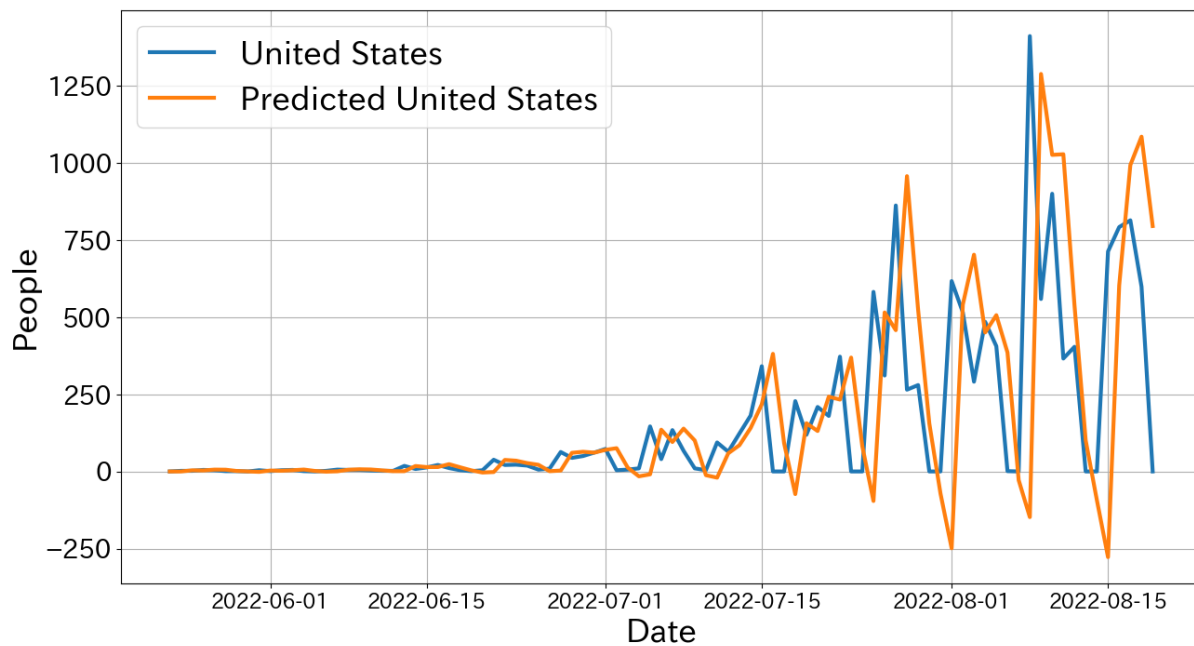


図 4.17: サル天然痘のデータを訓練させた ARIMA

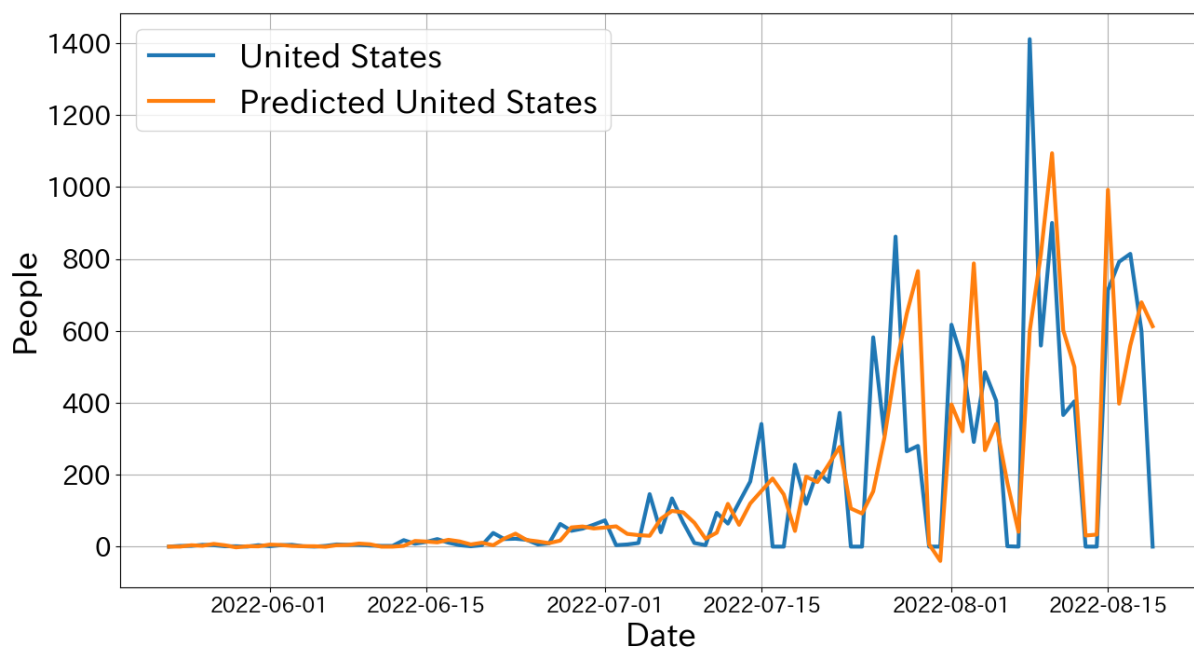


図 4.18: サル天然痘のデータを訓練させた SARIMA

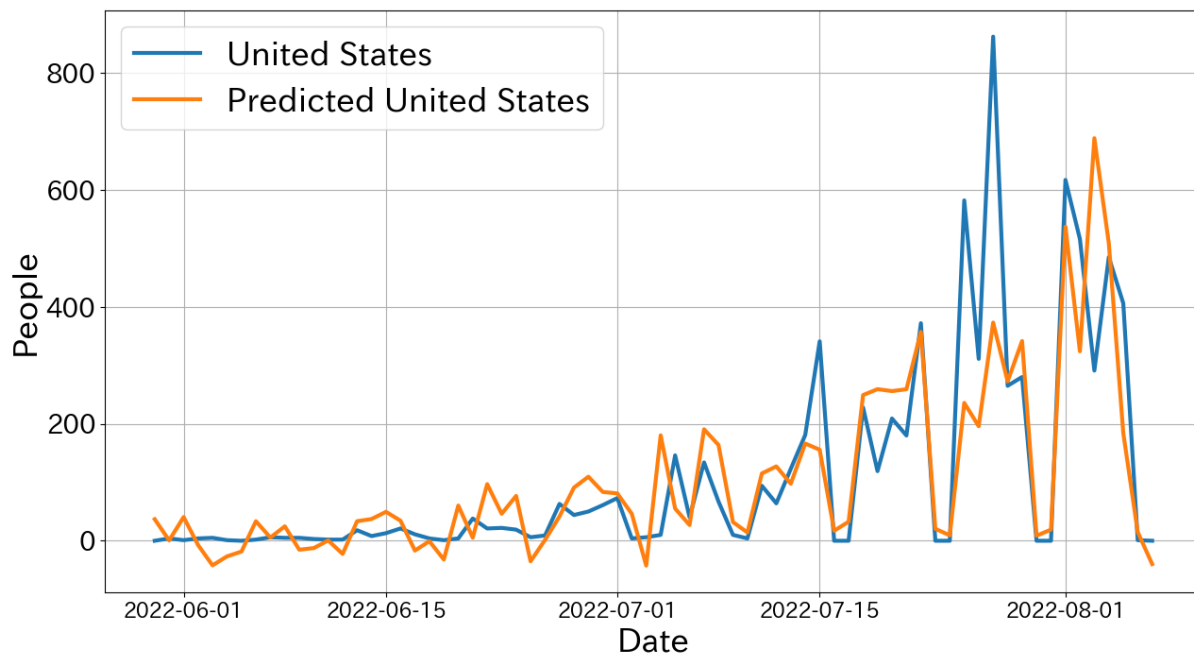


図 4.19: サル天然痘のデータを訓練させた RNN

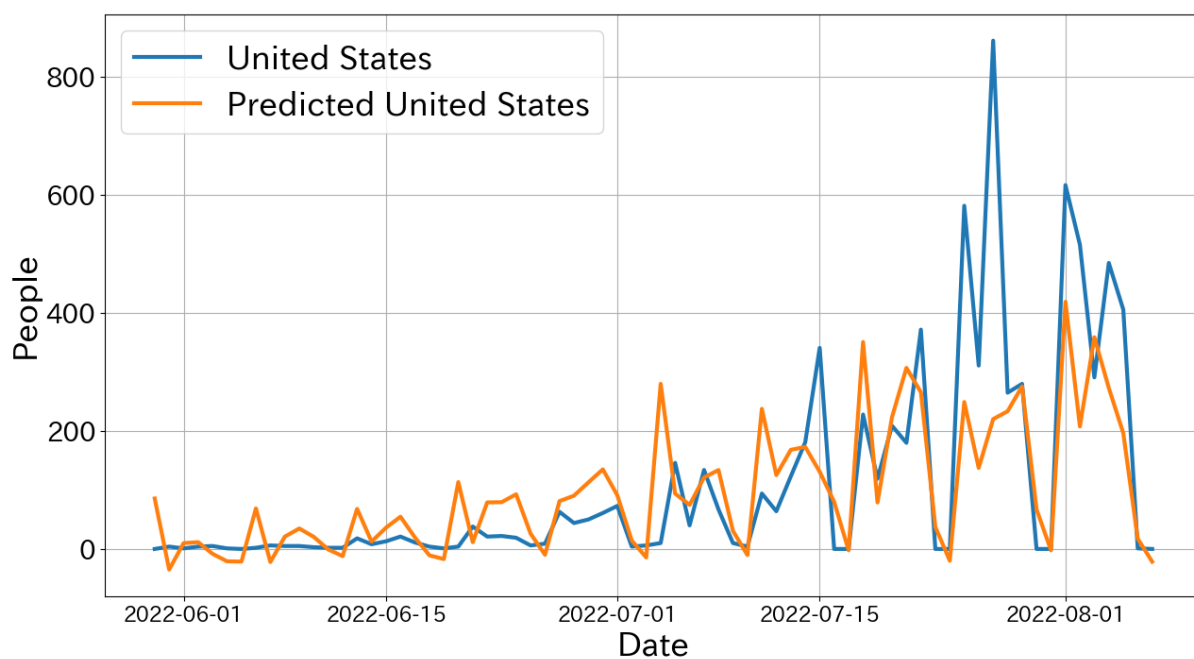


図 4.20: サル天然痘のデータを訓練させた LSTM

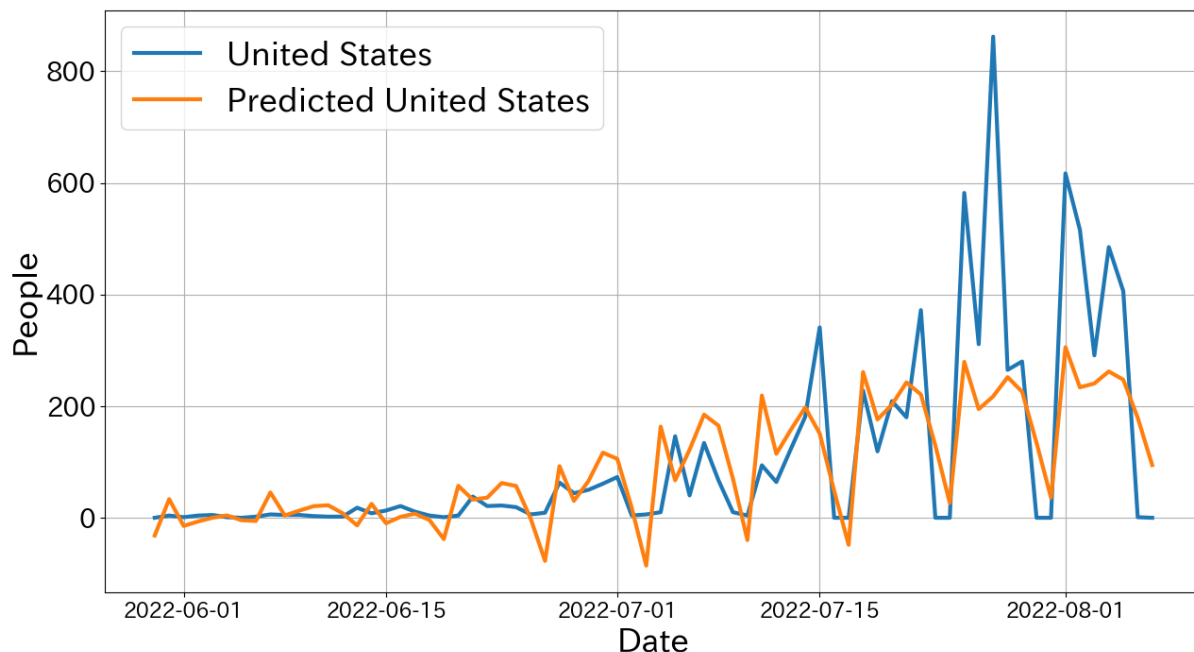


図 4.21: サル天然痘のデータを訓練させた Seq2Seq

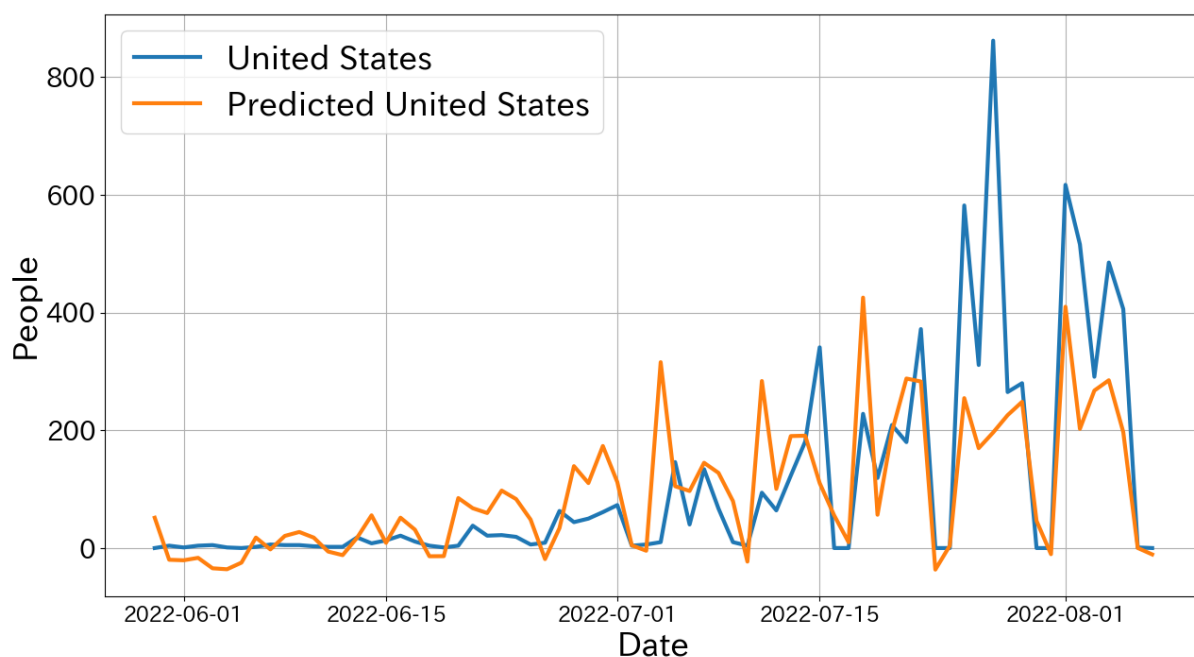


図 4.22: サル天然痘のデータを訓練させた StackedLSTM

図 4.19 から図 4.22 より、再帰型のニューラルネットワークによる予測結果はどのモデルも 0 の値が多い時系列の初期の区間において、予測値が上下している。また、時系列の後半に存在する 800 人を超えるピーク波形を正確に予測できていない。しかし、800 人を超えるピーク波形以外は周期性を捉えている。

次に、統計モデル (ARIMA、SARIMA) 及び 3 次元入力 3 次元出力の再帰型ニューラルネットワークモデル (RNN、LSTM、Seq2Seq、Stacked LSTM) によって、モデル構

築に用いなかったテスト区間の予測を行った。そして、予測値と実際の観測値との誤差をRMSEで評価した結果を表4.7に与える。なお、表4.7中のStackedはStacked LSTMを意味する。また、表4.7の差(%)は、サル痘のデータを用いて学習したARIMAのRMSEを基準として何%誤差が増減したかを意味する。差(%)は、式3.2で算出された。

表4.7: サル痘のデータを訓練させたモデルによるテスト区間の症例数予測結果の評価

	ARIMA	SARIMA	RNN	LSTM	Seq2Seq	Stacked
RMSE(人)	1629	276	360	450	474	523
差(%)	0.0	-83.1	-77.9	-72.4	-70.9	-67.9

表4.7より、統計モデル(ARIMA、SARIMA)の中ではSARIMAの方がテスト区間の予測誤差を最小化できていた。3次元入力3次元出力の再帰型のニューラルネットワークの中では、Stacked LSTM > Seq2Seq > LSTM > RNNの順にRMSEが小さかった。全モデルの中で、最も小さな予測誤差を示したのは、SARIMAであった。また表4.7中の差(%)より、統計モデルのARIMAのRMSEとの相対的な予測誤差(RMSE)の増減率においても、最も減少したモデルはSARIMAであった。再帰型のニューラルネットワークにおいて、Seq2Seq及びStacked LSTMモデルは単層のRNNや単層のLSTMよりも精度が高くなることを期待していたが、単層のRNNと単層のLSTMの方がRMSEが小さかった。ただし、どちらのモデルとも統計モデルのSARIMAと比べた際はRMSEをARIMAのRMSEと比べて5%以上減らす事ができていなかった。

次に、表4.7に与えた本研究の実験3の結果と表4.4のWuらの実験結果を比較した。Wuらのモデルは、インフルエンザ様疾患の前週の発生数との比率のデータで構築されている。その為、単純比較はできない。しかし、実験3で作成したモデルの中で最も予測誤差の小さいSARIMAモデルでは、Wuらの構築したTransformerよりも約40.7%高い精度を示した。また、4種類の再帰型のニューラルネットワークの中で最も予測誤差の小さいRNNでは、Wuらの構築したTransformerよりも約35.5%高い精度を示した。

#### 4.5.2. 考察

表4.7の結果について考察する。統計モデルにおいて、SARIMAがARIMAに対して約83.1%RMSEを減少した。その要因は、モデルに含まれる季節性の有無だと考える。以下図4.23に、ARIMA及びSARIMAによってテスト区間を予測させた結果を与える。

図4.23aより、ARIMAの予測は右肩下がりの予測になっている。これは、学習区間の後半の時系列の波の収束時のトレンドをARIMAが学習した為だと考える。一方で、図4.23bのSARIMAの予測結果は1週間周期の振幅を追従できている。

ここで図4.24に、ARIMA及びSARIMAに訓練させた区間予測させた予測値と観測値の残差の自己相関関数のコレログラムを与える。コレログラムは、各ラグにおける自己相関をプロットした図のことを言う。図4.7に示した青色の区間は、95%の信頼区間でこの区間を超えた自己相関は統計的に有意な自己相関とされる。

図4.24aより、ARIMAの残差には7日間周期の季節成分が含まれており、モデルに季節的な周期成分が含むことができていないと考えられる。また、図4.24aの結果から、SARIMAの残差では、自己相関がほとんど無くモデルに季節成分を取り込んでいると考

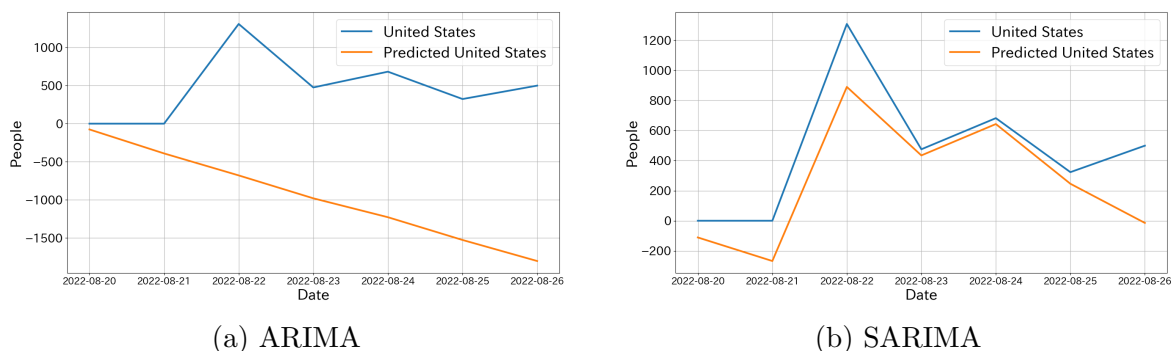


図 4.23: サル痘のデータを訓練させた統計モデルによるテスト区間の予測結果

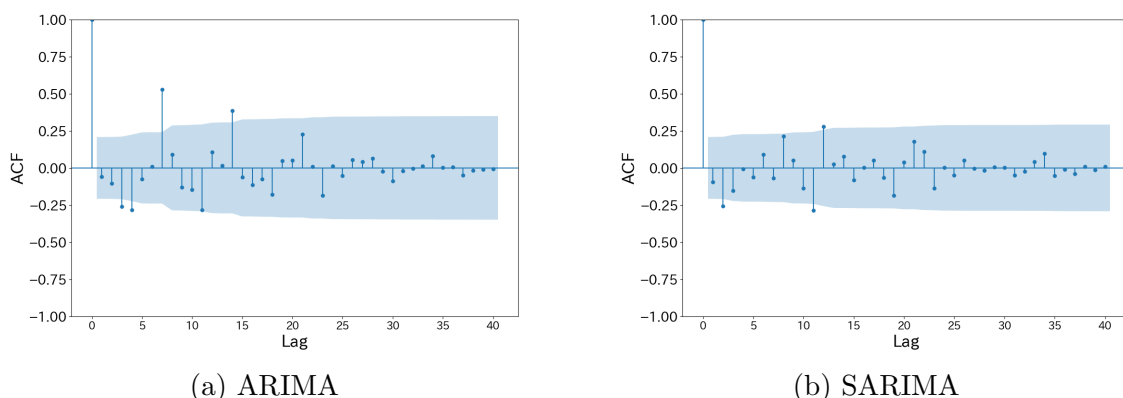


図 4.24: サル痘のデータを訓練させたモデルの残差における自己相関関数のコレログラム

えられる。このことから、ARIMA はトレンド成分のみを学習し、SARIMA はトレンドに加えて季節成分を学習できていると言える。

次に、4種類の再帰型のニューラルネットワークモデル (RNN、LSTM、Seq2Seq、Stacked LSTM) にテストデータを入力し、予測した結果について考察する。

表 4.7 から、4種類の再帰型のニューラルネットワークモデル (RNN、LSTM、Seq2Seq、Stacked LSTM) の中で最も予測精度が高かったのは RNN である事がわかる。また、図 4.19 から図 4.22 においても RNN のフィッティングが最も良いことがわかる。

これは、データセットの量が関係していると考えられる。Bashar ら [45] は、LSTM はパラメータ数が多いことから膨大な数のデータセットで学習させる必要があり、女性差別的発言をクラスタリングする際のデータセットなどでそのサイズが小さい場合モデルパラメータは過剰にデータセットに適応すると述べている。その為、今回のサル痘症例数のようにまだデータの観測が始まったばかりでデータセットの量が十分でない場合はパラメータの多い LSTM やその派生モデルの Seq2Seq や Stacked LSTM は予測モデルとして選択することに不適だと考える。その為、今回のサル痘症例数データにおいては LSTM やその派生モデルでは学習が不十分であり、その結果相対的に RNN の精度が高かったと考える。

今回作成した SARIMA 及び 4種類の再帰型のニューラルネットワークモデル (RNN、LSTM、Seq2Seq、Stacked LSTM) は、全て Wu ら [9] のインフルエンザ予測用 Transformer モデルの精度を大きく超える結果となった。この理由として、ARIMA が収束時の下降トレンドを学習していた為予測値が大きく外れたことと SARIMA や 4種類の再帰型のニューラルネットワークモデルがデータの周期性を追従できた為と考える。なお、SARIMA と RNN

を比較した際に SARIMA の方がテスト区間の RMSE が小さかった。これは、SARIMA の推定にカルマンフィルター [46] を用いたことで感染状況における状態をより良く推定できたからと考察する。



## 第5章 結論

本研究では、様々な感染症に対応できる時系列モデルの構築のため COVID-19 及びサル痘の症例数データを用いて実験1から実験3を行った。

本研究で使用する再帰型のニューラルネットワークはさまざまな入出力関係を持つことが可能である。そこで、実験1では以降の実験で使用する入出力関係を決定する為3種類の入出力関係のRNNを構築し統計モデルのARIMA及びSARIMAとの比較を行った。モデルには、COVID-19の第2波までのデータを各モデルに与え最後の1週間の症例数を予測した。また、Wuら[9]の研究と比較するためARIMAのRMSEを基準とした誤差の増減率を算出し比較を行った。結果として、Wuらの実験条件と異なるため単純比較はできないが、全てのモデルでWuらのTransformerより約33%以上高い精度を示した。特に、3次元入力3次元出力のRNNでは約47.5%高い精度を示した。この精度の理由として、多次元入出力のMany to Many RNNにおいて中間層の活性化関数であるtanh関数が自己系列に対するAdditive-Attention[44]として機能したと考えた。

実験2及び実験3では、実験1の結果を基にRNNとその派生モデルのLSTM、Stacked LSTM、Seq2Seqを3次元入力3次元出力のMany to Manyモデルとして構築した。

実験2では、各モデルに第7波までのCOVID-19のデータを与え訓練し評価した。更に、Wuら[9]の研究との比較も行った。Wuらの研究で最高精度のTransformerよりも、我々の統計モデルで最も精度の高いSARIMAでは約33.8%、我々の再帰型のニューラルネットワークで最も精度の高いLSTMでは約19.2%高い精度を示した。

実験3では、各モデルにサル痘に関するデータを与え訓練し評価した。更に、Wuら[9]の研究との比較も行った。Wuらの研究で最高精度のTransformerよりも、我々の統計モデルで最も精度の高いSARIMAでは約40.7%、我々の再帰型のニューラルネットワークで最も精度の高いRNNでは約35.5%高い精度を示した。

上述の結果をまとめると、感染症予測においてSARIMAモデルの予測精度が最も高い。機械学習を用いた場合は、数ヶ月程度の小規模のデータセットでの訓練ではRNNの精度が、1から2年程度の中規模のデータセットでの訓練ではLSTMの精度が高いといえる。

また、実験2及び実験3を通じて最も高い精度を示したSARIMAにおいて、表4.2と表4.6bより次数探索結果の上位3つのモデルで次数が共通していた。これは、少なくともCOVID-19とサル痘の間には感染を発生させる原因に共通点があることを示している。この共通の要因には、両実験においてSARIMAが最も高い精度を示していたことより季節性が含まれていると考えている。そのため、今回SARIMAの精度には及ばなかったが高精度を示していた単層のRNNやLSTMにおいても今後残差のコレログラムを描画しその季節性の有無を解析する事でより精度の高いモデル構築へ繋がると考えられる。季節性を上手く取り込む為には、階差データを用いた学習が必要だと考えられる。Hewamalageらは、季節的パターンがありデータ数が十分な際にRNN(LSTM)はモデリング可能であり、季節パターンが存在しない場合はそれらを作成する前処理が必要だと述べている[47]。

今回の COVID-19 及びサル痘のデータセットは、両方とも時刻によって振幅が大きく異なるデータである。そのため、モデリングが難しいと考える。そこで、1 次階差を取る前処理を行うことで非定常データを定常データに変換することができ、モデリングが容易になると考えられる。従って、今後の活動では残差のコレログラムにおける季節性の確認を行い階差データを用いた際のモデルの振る舞いを確認していく。

また、全実験を通じて自己系列に対する Additive-Attention[44] と考えられるモデルの振る舞いが確認された。そこで、他のデータセットを用いて自己系列に対する Additive-Attention として機能しているか否かの検証を行う必要がある。従って、今後の活動ではモデルの重みを解析を行なっていく。

さらに、直近の研究では Hasani らとその関連研究によって [48, 49, 50, 51, 52] によって、RNN に常微分方程式を導入することでより高い精度で入力の変化にも対応するモデルが構築されている。このモデルでは、今まで離散値しか取り扱うことができなかつた RNN に微分方程式を導入することで連続値を扱うことができる様になっている。また、Salinas ら [53] は、対数尤度を損失関数とすることで予測の際に誤差の 95% の信頼区間も示している。我々は、今後の活動でこのモデルのように対数尤度を導入することで関数の近似を行っていた RNN が時系列の発生させる分布を近似できる様になることを期待している。

今後の研究では、これらのモデルも含めて検証することでより良い感染症予測モデル構築への貢献に繋がることを期待し結論とする。

# 謝辞

本研究を遂行するにあたり、ご多忙の中数々のご指導及び助言をご教授していただきました高知工科大学システム工学群電子・光システム工学教室星野孝総准教授に深く心から感謝いたします。兼ねては、本システム開発に置きまして高知工科大学システム工学群 Soft Intelligent System on a Chip 研究室及び同研究室の OB・OG の皆様にさまざまな相談を快く引き受けてくださったことを感謝いたします。

最後になりましたが、学部4年間及び大学院2年間の生活を支えてくれた両親及び祖母に深く感謝致します。

## 参考文献

- [1] Hangargekar, CB, Quazi, RS, Joshi, AA: "A review on COVID-19-a global battle between life and death", *International Journal of Current Pharmaceutical Research*, pp. 19–24, 2020.
- [2] "The species Severe acute respiratory syndrome-related coronavirus: classifying 2019-nCoV and naming it SARS-CoV-2", *Nature microbiology*, Vol. 5, No. 4, pp. 536–544, 2020.
- [3] Wrapp, Daniel, Wang, Nianshuang, Corbett, Kizzmekia S, Goldsmith, Jory A, Hsieh, Ching-Lin, Abiona, Olubukola, Graham, Barney S, McLellan, Jason S: "Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation", *Science*, Vol. 367, No. 6483, pp. 1260–1263, 2020.
- [4] Organization, World Health, et al.: "Weekly epidemiological update on COVID-19 - 14 December 2022", 2022.
- [5] 厚生労働省: "第112回新型コロナウイルス感染症対策アドバイザリーボード", p. 1, 2022.
- [6] Magnus, Preben von, Andersen, Else Krag, Petersen, Knud Birkum, Birch-Andersen, Aksel: "A pox-like disease in cynomolgus monkeys", *Acta Pathologica Microbiologica Scandinavica*, Vol. 46, No. 2, pp. 156–176, 1959.
- [7] Arita, I, Henderson, DA: "Smallpox and monkeypox in non-human primates", *Bulletin of the World Health Organization*, Vol. 39, No. 2, p. 277, 1968.
- [8] 厚生労働省健康局結核感染症課: "サル痘に関する情報提供及び協力依頼について", 20/05/2022.
- [9] Wu, Neo, Green, Bradley, Ben, Xue, O'Banion, Shawn: "Deep transformer models for time series forecasting: The influenza prevalence case", *arXiv preprint arXiv:2001.08317*, 2020.
- [10] Bartholomew, David J: "Time Series Analysis Forecasting and Control.", pp. 199–201, 1971.
- [11] Hochreiter, Sepp, Schmidhuber, Jürgen: "Long short-term memory", *Neural computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

- [12] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, Polosukhin, Illia: "Attention is all you need", *Advances in neural information processing systems*, Vol. 30, , 2017.
- [13] Wolf, Thomas, Debut, Lysandre, Sanh, Victor, Chaumond, Julien, Delangue, Clement, Moi, Anthony, Cistac, Pierric, Rault, Tim, Louf, Rémi, Funtowicz, Morgan, et al.: "Transformers: State-of-the-art natural language processing", pp. 38–45, 2020.
- [14] Jamil, Sonain, Piran, Md, Kwon, Oh-Jin, et al.: "A Comprehensive Survey of Transformers for Computer Vision", *arXiv preprint arXiv:2211.06004*, 2022.
- [15] Li, Shiyang, Jin, Xiaoyong, Xuan, Yao, Zhou, Xiyou, Chen, Wenhui, Wang, Yuxiang, Yan, Xifeng: "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting", *Advances in neural information processing systems*, Vol. 32, , 2019.
- [16] Tang, Binh, Matteson, David S: "Probabilistic transformer for time series analysis", *Advances in Neural Information Processing Systems*, Vol. 34, pp. 23592–23608, 2021.
- [17] Zerveas, George, Jayaraman, Srideepika, Patel, Dhaval, Bhamidipaty, Anuradha, Eickhoff, Carsten: "A transformer-based framework for multivariate time series representation learning", *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2114–2124, 2021.
- [18] Dosovitskiy, Alexey, Beyer, Lucas, Kolesnikov, Alexander, Weissenborn, Dirk, Zhai, Xiaohua, Unterthiner, Thomas, Dehghani, Mostafa, Minderer, Matthias, Heigold, Georg, Gelly, Sylvain, et al.: "An image is worth 16x16 words: Transformers for image recognition at scale", *arXiv preprint arXiv:2010.11929*, 2020.
- [19] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, Toutanova, Kristina: "Bert: Pre-training of deep bidirectional transformers for language understanding", *arXiv preprint arXiv:1810.04805*, 2018.
- [20] Zhou, Haoyi, Zhang, Shanghang, Peng, Jieqi, Zhang, Shuai, Li, Jianxin, Xiong, Hui, Zhang, Wancai: "Informer: Beyond efficient transformer for long sequence time-series forecasting", Vol. 35, No. 12, pp. 11106–11115, 2021.
- [21] Lee, Seung Hoon, Lee, Seunghyun, Song, Byung Cheol: "Vision transformer for small-size datasets", *arXiv preprint arXiv:2112.13492*, 2021.
- [22] ArunKumar, KE, Kalaga, Dinesh V, Kumar, Ch Mohan Sai, Kawaji, Masahiro, Brenza, Timothy M: "Comparative analysis of Gated Recurrent Units (GRU), long Short-Term memory (LSTM) cells, autoregressive Integrated moving average (ARIMA), seasonal autoregressive Integrated moving average (SARIMA) for forecasting COVID-19 trends", *Alexandria Engineering Journal*, Vol. 61, No. 10, pp. 7585–7603, 2022.

- 
- [23] Makridakis, Spyros, Spiliotis, Evangelos, Assimakopoulos, Vassilios: "Statistical and Machine Learning forecasting methods: Concerns and ways forward", *PloS one*, Vol. 13, No. 3, p. e0194889, 2018.
- [24] Rosenblatt, Frank: "The perceptron: a probabilistic model for information storage and organization in the brain.", *Psychological review*, Vol. 65, No. 6, p. 386, 1958.
- [25] Elman, Jeffrey L: "Finding structure in time", *Cognitive science*, Vol. 14, No. 2, pp. 179–211, 1990.
- [26] Jordan, Michael I: "Serial order: A parallel distributed processing approach", Vol. 121, pp. 471–495, 1997.
- [27] Rumelhart, David E, Hinton, Geoffrey E, Williams, Ronald J: "Learning representations by back-propagating errors", *nature*, Vol. 323, No. 6088, pp. 533–536, 1986.
- [28] Deng, Li: "A tutorial survey of architectures, algorithms, and applications for deep learning", *APSIPA transactions on Signal and Information Processing*, Vol. 3, , 2014.
- [29] Werbos, Paul J: "Backpropagation through time: what it does and how to do it", *Proceedings of the IEEE*, Vol. 78, No. 10, pp. 1550–1560, 1990.
- [30] Pascanu, Razvan, Mikolov, Tomas, Bengio, Yoshua: "On the difficulty of training recurrent neural networks", *International conference on machine learning*, pp. 1310–1318. PMLR, 2013.
- [31] Ribeiro, Antônio H, Tiels, Koen, Aguirre, Luis A, Schön, Thomas: "Beyond exploding and vanishing gradients: analysing RNN training using attractors and smoothness", *International Conference on Artificial Intelligence and Statistics*, pp. 2370–2380. PMLR, 2020.
- [32] Gers, Felix A, Schmidhuber, Jürgen, Cummins, Fred: "Learning to forget: Continual prediction with LSTM", 1999.
- [33] Maaliw, Renato R, Mabunga, Zoren P, Villa, Frederick T: "Time-Series Forecasting of COVID-19 Cases Using Stacked Long Short-Term Memory Networks", pp. 435–441, 2021.
- [34] Sutskever, Ilya, Vinyals, Oriol, Le, Quoc V: "Sequence to sequence learning with neural networks", *Advances in neural information processing systems*, Vol. 27, , 2014.
- [35] Abbasimehr, Hossein, Paki, Reza: "Improving time series forecasting using LSTM and attention models", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 13, No. 1, pp. 673–691, 2022.
- [36] Seabold, Skipper, Perktold, Josef: "statsmodels: Econometric and statistical modeling with python", 2010.
- [37] Chollet, François, et al.: "Keras", 2015.

- [38] Abadi, Martín, Agarwal, Ashish, Barham, Paul, Brevdo, Eugene, Chen, Zhifeng, Citro, Craig, Corrado, Greg S., Davis, Andy, Dean, Jeffrey, Devin, Matthieu, Ghemawat, Sanjay, Goodfellow, Ian, Harp, Andrew, Irving, Geoffrey, Isard, Michael, Jia, Yangqing, Jozefowicz, Rafal, Kaiser, Lukasz, Kudlur, Manjunath, Levenberg, Josh, Mané, Dandelion, Monga, Rajat, Moore, Sherry, Murray, Derek, Olah, Chris, Schuster, Mike, Shlens, Jonathon, Steiner, Benoit, Sutskever, Ilya, Talwar, Kunal, Tucker, Paul, Vanhoucke, Vincent, Vasudevan, Vijay, Viégas, Fernanda, Vinyals, Oriol, Warden, Pete, Wattenberg, Martin, Wicke, Martin, Yu, Yuan, Zheng, Xiaoqiang: "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems", 2015. Software available from tensorflow.org.
- [39] Ozaki, T: "On the order determination of ARIMA models", *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, Vol. 26, No. 3, pp. 290–301, 1977.
- [40] "<https://github.com/CSSEGISandData/COVID-19>".
- [41] Shahid, Farah, Zameer, Aneela, Muneeb, Muhammad: "Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM".
- [42] Kermack, William Ogilvy, McKendrick, Anderson G: "A contribution to the mathematical theory of epidemics", *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, Vol. 115, No. 772, pp. 700–721, 1927.
- [43] "<https://www.mhlw.go.jp/stf/covid-19/open-data.html>".
- [44] Bahdanau, Dzmitry, Cho, Kyunghyun, Bengio, Yoshua: "Neural machine translation by jointly learning to align and translate", *arXiv preprint arXiv:1409.0473*, 2014.
- [45] Bashar, Md Abul, Nayak, Richi, Suzor, Nicolas: "Regularising LSTM classifier by transfer learning for detecting misogynistic tweets with small training set", *Knowledge and Information Systems*, Vol. 62, No. 10, pp. 4029–4054, 2020.
- [46] Kalman, Rudolph Emil: "A new approach to linear filtering and prediction problems", 1960.
- [47] Hewamalage, Hansika, Bergmeir, Christoph, Bandara, Kasun: "Recurrent neural networks for time series forecasting: Current status and future directions", *International Journal of Forecasting*, Vol. 37, No. 1, pp. 388–427, 2021.
- [48] Hasani, Ramin, Lechner, Mathias, Amini, Alexander, Rus, Daniela, Grosu, Radu: "Liquid time-constant networks", *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, pp. 7657–7666, 2021.
- [49] Hasani, Ramin, Lechner, Mathias, Amini, Alexander, Liebenwein, Lucas, Ray, Aaron, Tschaikowski, Max, Teschl, Gerald, Rus, Daniela: "Closed-form continuous-time neural networks", *Nature Machine Intelligence*, pp. 1–12, 2022.

- [50] Chen, Ricky TQ, Rubanova, Yulia, Bettencourt, Jesse, Duvenaud, David K: "Neural ordinary differential equations", *Advances in neural information processing systems*, Vol. 31, , 2018.
- [51] Funahashi, Ken-ichi, Nakamura, Yuichi: "Approximation of dynamical systems by continuous time recurrent neural networks", *Neural networks*, Vol. 6, No. 6, pp. 801–806, 1993.
- [52] Mozer, Michael C, Kazakov, Denis, Lindsey, Robert V: "Discrete event, continuous time rnns", *arXiv preprint arXiv:1710.04110*, 2017.
- [53] Salinas, David, Flunkert, Valentin, Gasthaus, Jan, Januschowski, Tim: "DeepAR: Probabilistic forecasting with autoregressive recurrent networks", *International Journal of Forecasting*, Vol. 36, No. 3, pp. 1181–1191, 2020.



# 研究業績

1. 02, 2021. ”新型コロナウイルス感染者数予測モデルの性能比較と評価” 日本知能情報ファジィ学会 中国・四国支部大会 (Online). 2021.02. 土肥発表
2. 03,2021.”新型コロナウイルス感染者数予測モデルの性能比較と評価” 第18回ポトラック&ワークショップ (Online). 2021.03. 土肥発表
3. 09, 2021. ”新型コロナウイルス確認症例数予測におけるニューラルネットワークと統計モデルの有用性の検証” Fuzzy System Symposium (FSS) 2021 (Online). 2021.09. 土肥発表
4. 10, 2021. ”A Comparative Study on Statistical Method and Neural Network in COVID-19 Forecasting” The 7th International Workshop on Advanced Computational Intelligence and Intelligent Informatics (IWACIII) 2021 (Online). 2021.10. 土肥発表
5. 03, 2022. ”Recurrent Neural Network を用いた新型コロナウイルス感染者数予測の検証” 令和3年度 第26回日本知能情報ファジィ学会中国・四国支部大会 (Online). 2022.03. 土肥発表
6. 09, 2022. ”深層学習を用いた COVID-19 症例数予測モデル最適化の検証” Fuzzy System Symposium (FSS) 2022 (Online). 2022.09. 土肥発表
7. 09, 2022. ”A comparative study of statistical models and univariate and multivariate RNN for forecasting the number of COVID-19 cases” The 10th International Symposium on Computational Intelligence and Industrial Applications (ISCIIA) 2022 (Online). 2022.09. 土肥発表
8. 12, 2022.”A comparative study for COVID-19 cases forecasting with loss function as AIC and MSE in RNN family and ARIMA” 2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS) 2022 (Online). 2022.12. 土肥発表
9. 2,2023.”COVID-19 症例数予測における Deep Learning モデル適用の実験的考察”, 日本知能情報ファジィ学会誌 知能と情報 (2023) Vol35 No.1 pp.501-506 2023.2.