

修士論文

ゲート波形モニタリングによるパワーデバイスの劣化推定に向けた AI 的手法の研究 AI-based Approach for Estimating Degradation of Power Devices by Gate Waveform Monitoring

平成 31 年 1 月 31 日

指導教員 桜井 貴康 教授

東京大学大学院 工学系研究科
電気系工学専攻 融合情報コース

学籍番号 37-176505

羅 揚

目次

第1章 序論	3
1.1 研究の背景および目的	3
1.2 パワーデバイスの劣化	4
1.3 ゲートドライバ回路の多機能化.....	7
1.4 本論文の構成	7
第2章 ダブルパルススイッチングテスト	9
2.1 ダブルパルス回路.....	9
2.2 ダブルパルス回路の動作.....	9
2.2.1 一回目のターンオンからターンオフ時まで(第一パルス).....	9
2.2.2 ターンオフから二回目のターンオン時まで(第二パルス).....	11
2.2.3 パワーデバイスのターンオン特性	12
第3章 機械学習とニューラルネットワーク	14
3.1 学習の流れ/アルゴリズムの選択.....	14
3.2 学習データの特徴抽出アルゴリズム.....	15
3.2.1 主成分分析.....	15
3.2.2 線形判別分析	16
3.2.3 畳み込みニューラルネットワーク	17
3.3 学習データの分類アルゴリズム.....	17
3.3.1 k-最近傍法.....	17
3.3.2 人工ニューラルネットワーク	17
3.4 オープンソースライブラリ	20
3.4.1 scikit-learn	20
3.4.2 Keras.....	21
第4章 パワーデバイスのパラメータ推定	22
4.1 先行研究	22
4.1.1 IGBT の劣化判定	22
4.1.2 ジャンクション温度の推定	26
4.2 提案手法	26
第5章 学習データの取得	28
5.1 波形データの取得手順	28
5.1.1 LTspice シミュレーション.....	28
5.1.2 高電圧測定.....	29
5.2 取得した波形データの見たい目.....	31

5.2.1 シミュレーション波形の確認	31
5.2.1 高電圧測定波形の確認	31
第 6 章 実験結果.....	37
6.1 採用した三種類の学習フローの学習結果	38
6.1.1 PCA+kNN を用いた学習	38
6.1.2 LDA+kNN を用いた学習	44
6.1.3 CNN を用いた学習	47
6.2 A/D 変換器性能による学習結果への影響	49
6.3 高電圧測定データを用いた検証	53
6.4 実験結果のまとめと考察	54
第 7 章 結論.....	56
参考文献.....	57
本研究に関する発表.....	59
謝辞.....	60

第1章 序論

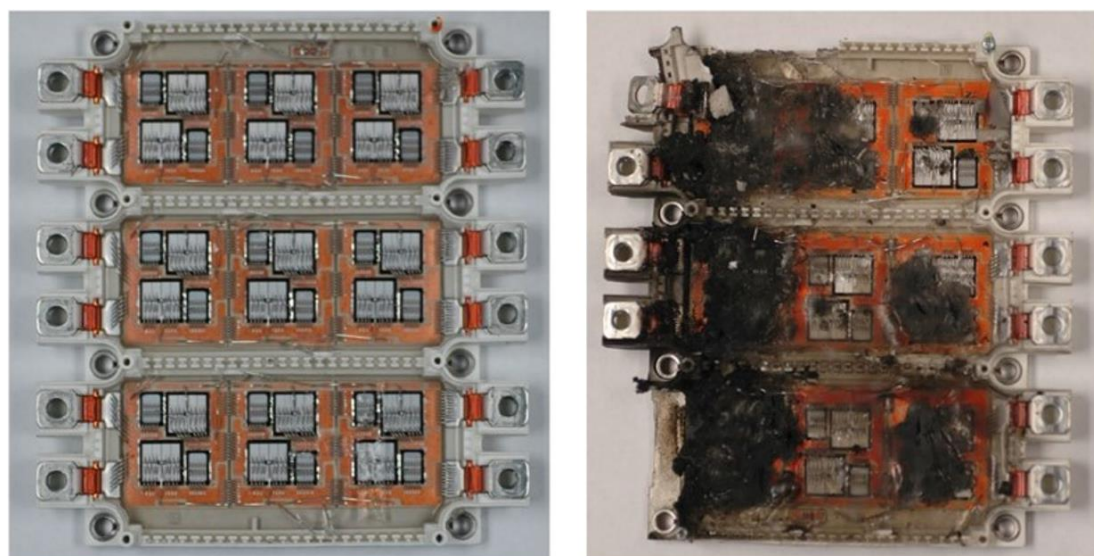
1.1 研究の背景および目的

本研究では、絶縁ゲートバイポーラトランジスタ(IGBT)やパワーMOSFETをはじめとする電力用半導体素子(以下、パワーデバイス)の劣化状況を推定するための手法を提案し、SPICE シミュレーションや実測テストを用いて取得した波形データを、機械学習およびニューラルネットワークモデルに学習させることによって、その有効性を検討した。

パワーデバイスは高電圧・大電流を扱えることを特徴とし、パワーエレクトロニクス分野において、電力制御用のスイッチとして幅広く使われている。具体的な応用例として、電力自動車・鉄道車両・風力発電機のシステムの一部であるモータ駆動装置や電力変換装置などにはパワーデバイスが含まれている。しかし、パワーデバイスは経年劣化によって生じたさまざまな不具合は、システム効率の低下およびシステムダウンに至るおそれがある。たとえば図 1-1 のように、IGBT モジュール経年劣化によって熱暴走を起こし、モジュールを燃やしてしまうケースが報告されている[1]。したがって、パワーデバイスの故障・不具合を早期警告するための手法が確立できれば、予定外のメンテナンスを防ぐとともに、システムの安全性を高めることにつながる。

パワーデバイスに起きる故障・不具合は、それらに關係する変化を観測することによって、一つまたは複数の兆候から予知することができる[2]。潜在的な故障・不具合を防止するための体系的な分析手法として、FMMEA(故障モードとそのメカニズムおよび影響の解析)が挙げられる[3]。兆候となるパラメータ(以下、デバイスパラメータ)を特定しモニタリングすることで、通常予想される動作条件からの逸脱や劣化の程度を評価することができる。結果的に、評価結果が故障・不具合に関するは判断材料となり、パワーデバイスの信頼性を向上させる。

デバイスパラメータをモニタリングするための方法論がいくつか提案されている。モニタリング対象をパワーデバイスのスイッチング回路における主回路(強電部)あるいは制御回路(弱点部)のいずれかから選択しているかによって大きく二分することができる。それぞれの例を挙げると、Patil ら異常検出手法を用いて、コレクタ-エミッタ間の電流値と電圧値から異常検出を行った[4]。一方、Baker らはパワーデバイスがターンオン時のゲート端子電圧のピーク値を測定することで、ジャンクション温度を推定した[5]。両者を比較すると、前者は従来の主回路からのセンシング手法を用いているため、



新品の I G B T

劣化によって熱暴走した I G B T

図 1-1 風力発電機用 IGBT モジュールの故障[1]

デバイスの状態は把握しやすく、複数の既知パラメータからなるデータセットを統計的に計算することで、ロバストなモニタリングを可能としている。後者の制御回路からのセンシング手法はデバイスの動作を把握することは一般的に難しいとされているが、ゲートの測定回路をゲートドライバに組み込めることができるため、高価な外付けの高電圧センシングを必要とせず、主回路の動作に影響を及ぼすこともないなどの利点が存在する。さらに、近年のゲートドライバ回路の高性能化・多機能化に伴い、制御回路からのモニタリング手法は注目されつつある。

本研究は将来的にゲートドライバに取り入れられることを視野に、ゲート波形を用いたモニタリング手法を提案する。複数のデバイスパラメータが同時に変動する中で取得されたゲート電圧のターンオン波形を学習データとし、機械学習やニューラルネットワークモデルに学習させる。パラメータ変動によって生じる、人間の目では見分けることが難しい微小な変化を高い正解率で読み取ることを目的とする。

1.2 パワーデバイスの劣化

日本工業規格(JIS)によれば、故障モードとは「故障状態の形式による分類。例えば、断線、短絡、折損、摩耗、特性の劣化など」と定義されている[6]。FMMEA は、潜在的な故障モードに対し、そのメカニズムを特定するための方法論である。FMMEA をパワーデバイスに対して行えば、デバイスの重大な不具合につながるメカニズムがリストア

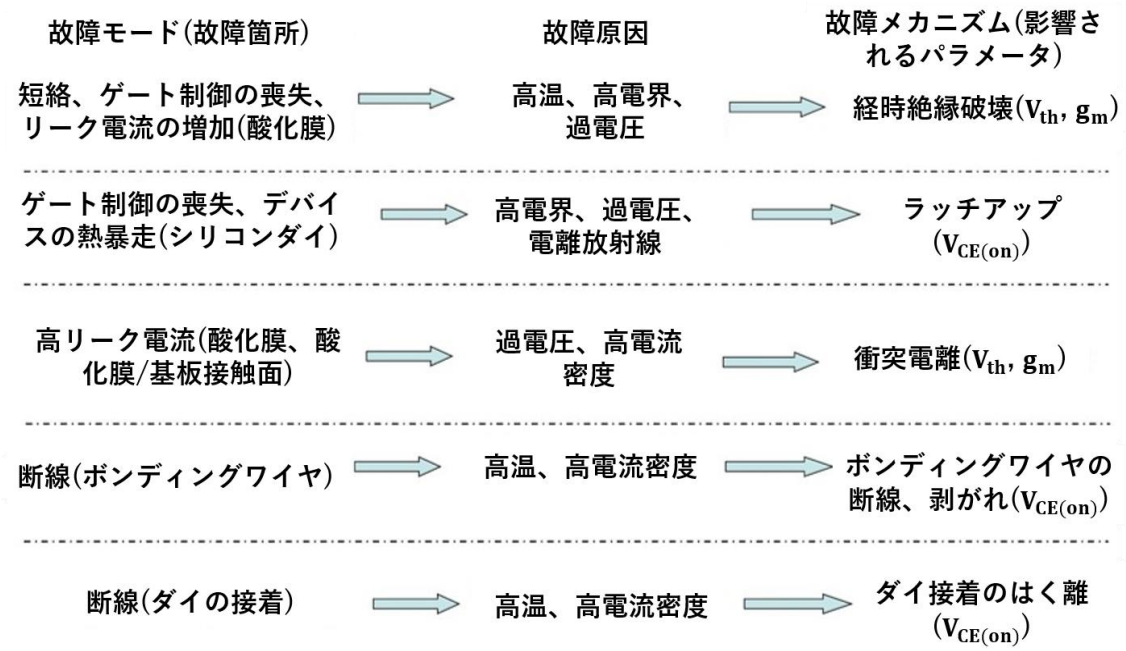


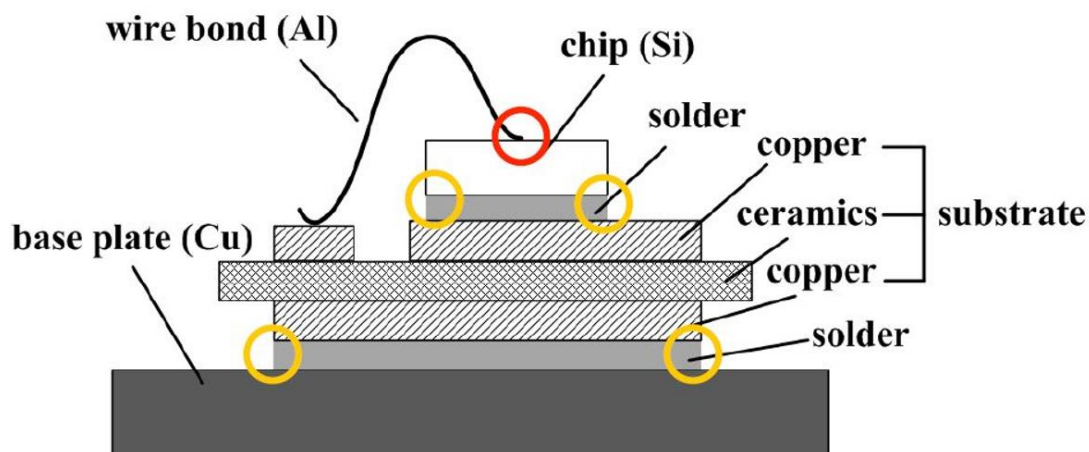
図 1-2 パワーデバイスモジュールの FMMEA [3]

ップされると同時に、そのメカニズムに関する物理現象やモニタリングすべきデバイスパラメータを明らかにすることができる。パワーデバイスモジュールの FMMEA で解析した結果は図 1-2 にまとめられている。

典型的なパワーデバイスモジュールは図 1-3 のように、半導体チップはダイレクトボンド銅(DBC)セラミック基板にはんだ付けられ、さらにこれを銅ベース板にはんだ付けしている。不具合が起き得る箇所として、チップのゲート酸化膜や、チップとセラミック基板、ボンディングワイヤ、ベース板間のはんだ接触面が挙げられる。したがって、パワーデバイスモジュールの故障モードには短絡、断続的な断線、リーク電流の増加、またはゲート制御の喪失(ターンオフできなくなる)などが含まれている。故障の原因は高温などの環境的な要因や熱・電氣的サイクルなどの動作的な要因からなるものであるが、パワーデバイスはこれらの条件下で繰り返し動作し続けると、疲労劣化現象が現れる。

デバイスの劣化による影響は、デバイスパラメータのモニタリングで観察することができる。また、故障モードによって、ある特定のデバイスパラメータは他のより大きく変化がみられることから、複数のデバイスパラメータをデバイス動作中に同時にモニタリングすることで、著しく変化がみられたパラメータからどの故障モードが起きたかを特定できるようになる。

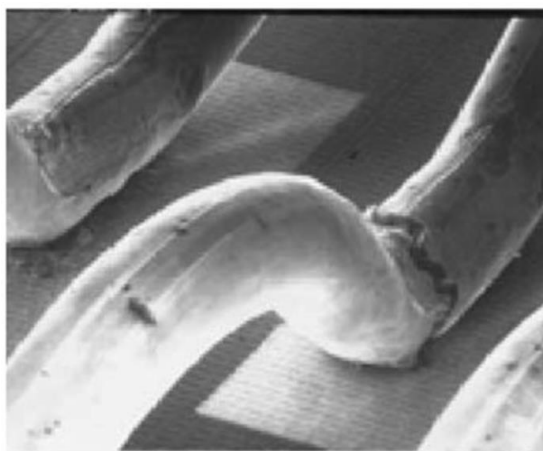
本研究はパワーデバイスモジュールのゲート酸化膜およびボンディングワイヤとチ



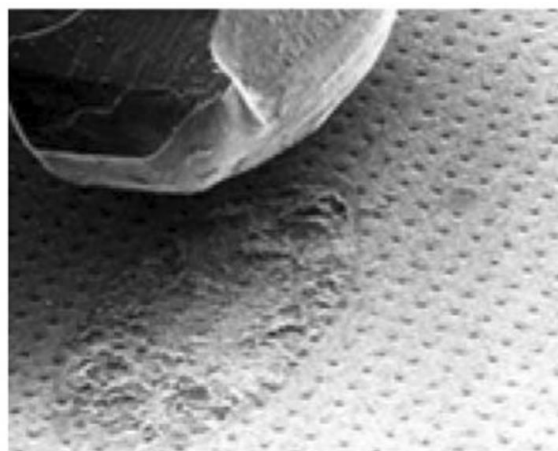
- ボンディングワイヤの劣化箇所

- はんだ接触面の劣化箇所

図 1-3 パワーデバイスモジュールの断面図[1]



ボンディングワイヤの断線



接触面からの剥がれ

図 1-4 ボンディングワイヤの劣化[1]

ップ間とのはんだ接触面の劣化に注目している。ゲート酸化膜の経時絶縁破壊(TDDB)現象は高温や高電界により衝突電離を引き起こし、生じたホット・エレクトロンがゲート酸化膜を劣化させ、デバイスの電圧しきい値(V_{th})が増加する。一方、ワイヤ、チップおよび基板にはそれぞれ異なる熱膨張率を持ち、この違いと熱・電氣的サイクルによる周期的な応力はワイヤに疲労劣化させ、結果的に図 1-4 のようなボンディングワイヤの断線や接触面からの剥がれを引き起こし、エミッタ/ソース端子における配線の抵抗

値(R_e/R_s)が徐々に増加する。

この二つの劣化を示すデバイスパラメータに動作条件であるジャンクション温度(T_j)および主回路に流れるコレクタ/ドレイン電流(I_c/I_d)を加え、合計四つのデバイスパラメータを推定対象とする。

1.3 ゲートドライバ回路の多機能化

シリコンを用いた半導体技術は成熟し、パワーデバイスの物理性能も理論的な限界値に達しているといわれている。そこで、システム全体の性能をさらに向上させるために、ゲートドライバの設計が注目され、さまざまな機能をゲートドライバ回路に組み込むことが提案されているようになった。このような多機能なゲートドライバを「スマートゲートドライバ」と呼ばれ、提案された機能群は図 1-5 に示すように大きく三種類に分かれている。[7]

一つ目は、パワーデバイスのスイッチング動作を最適化するための機能群である。例えば、ターンオン・ターンオフ時の電圧・電流波形を所望な形にするアクティブゲート制御は、スイッチング速度やスイッチング損失、電磁波障害(EMI)間のトレードオフを調節することを目的としている。これらは最適なスイッチング特性を得るのにもっとも考慮すべき要素である。

二つ目は、短絡保護や過電流保護、過熱保護、過電圧/不足電圧保護など、パワーデバイスにさまざまな保護を施すための機能群である。これらはデバイスの信頼性と安全性を確保するのに必須であり、近年はよりロバストかつ短い反応速度を実現できるような保護手法に着目している。

三つ目は、温度検出や電流検出、電流調節、電流の平衡化などをはじめとするモニタリングと調整機能群である。これらはデバイスの機能性や順応性、信頼性を向上するための比較的新しいアプローチであり、本研究で扱っている背景である。

1.4 本論文の構成

本論文は7章で構成されている。第2章では、パワーデバイスの動作を模擬するダブルパルススイッチングテストについて解説する。第3章では、機械学習とニューラルネットワークのオープンソースライブラリを紹介し、本研究で用いた学習データの特徴抽出手法のアルゴリズムをまとめる。第4章では、デバイスパラメータの推定手法を提案し、先行研究との比較を行う。第5章では、学習に使う実測とシミュレーションデータの取得方法、および実装する際にA/D変換回路の性能による影響の再現について説明する。第6章では、各学習アルゴリズムで学習した結果を示し、その考察

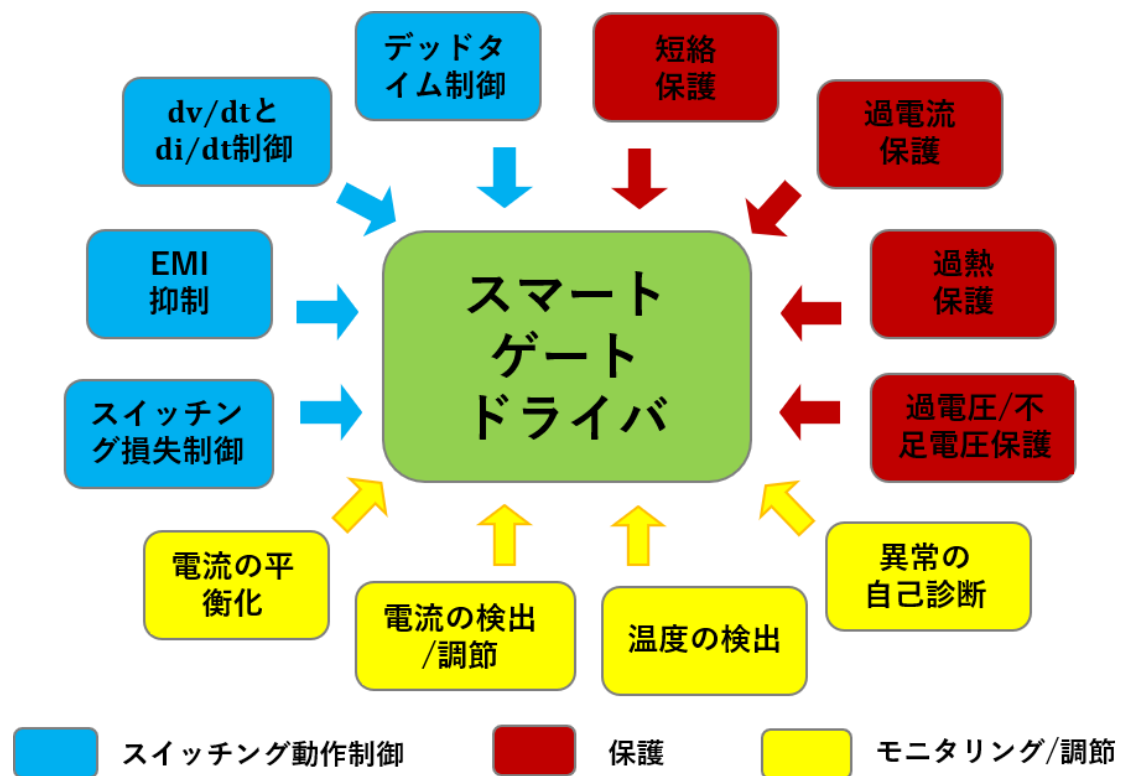


図 1-5 スマートゲートドライバに提案されたさまざまな機能[7]

を行う。最後に第7章で本研究のまとめを述べる。

第2章 ダブルパルススイッチングテスト

2.1 ダブルパルス回路

ダブルパルス回路は、インバータ実機に組み込まれたパワーデバイスのスイッチング性能を確認するために置き換えられた簡易的なテスト回路である[8]。

パワーデバイスのスイッチング動作は、図 2-1 に示す主なデバイス端子間の寄生容量、すなわちゲート-コレクタ/ドレイン間容量($C_{GC/GD}$)、ゲート-エミッタ/ソース間容量($C_{GE/GS}$)、コレクタ/ドレイン-エミッタ/ソース間容量($C_{CE/DS}$)に左右される。デバイスのターンオン時およびターンオフ時の寄生容量における充放電はゲート電圧の時定数を影響し、最終的にデバイスのスイッチング速度の上限が決定される。

ダブルパルス回路の等価回路を図 2-2 に示す。この回路は、インバータの一相分を取り出した形になっているため、インバータ動作時と同等な条件でパワーデバイスのスイッチング特性を見ることができる。任意の I_c/I_d におけるターンオン特性とターンオフ特性を観測するには、電流を環流させる必要がある。そのため、パワーデバイスの上側に環流ダイオードと環流インダクタが逆並列に接続されている。

2.2 ダブルパルス回路の動作

ダブルパルス回路の動作は以下の通りである。主回路に高電圧の DC 電源を与え、試験用デバイスのゲート端子に所望の I_c/I_d になるように電圧パルスの幅(ターンオン・ターンオフのタイミング)を印加する。図 2-3 にダブルパルス回路の動作原理を示す。

2.2.1 一回目のターンオンからターンオフ時まで(第一パルス)

通常ダブルパルス回路は電流が 0 の状態から起動し、図 2-4 のターンオン状態になる。このとき、電流はパワーデバイスを介してインダクタに

$$\frac{di}{dt} = \frac{V_{DD}}{L} \quad (1)$$

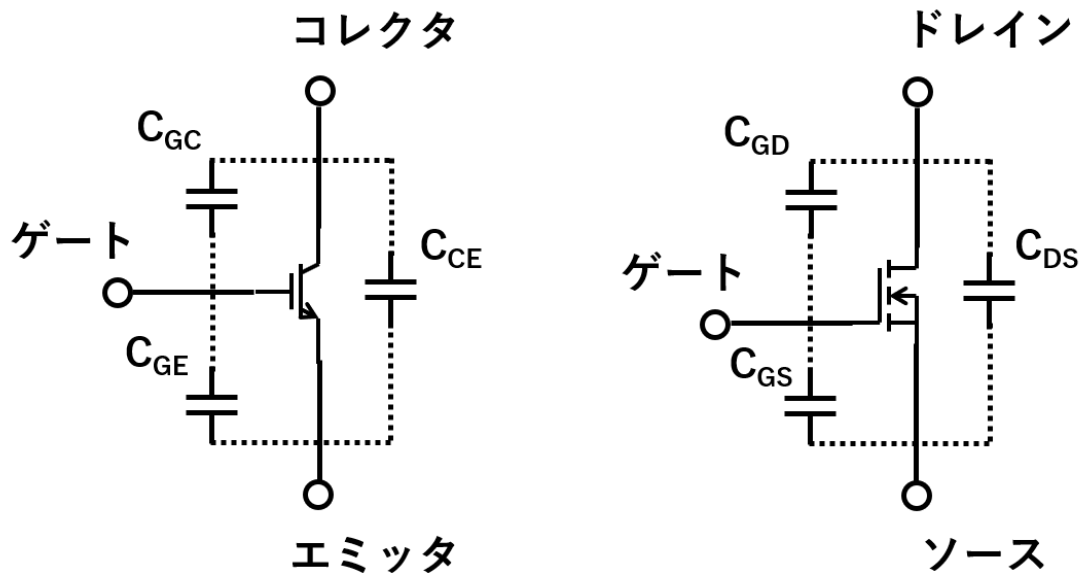


図 2-1 IGBT と MOSFET の寄生容量

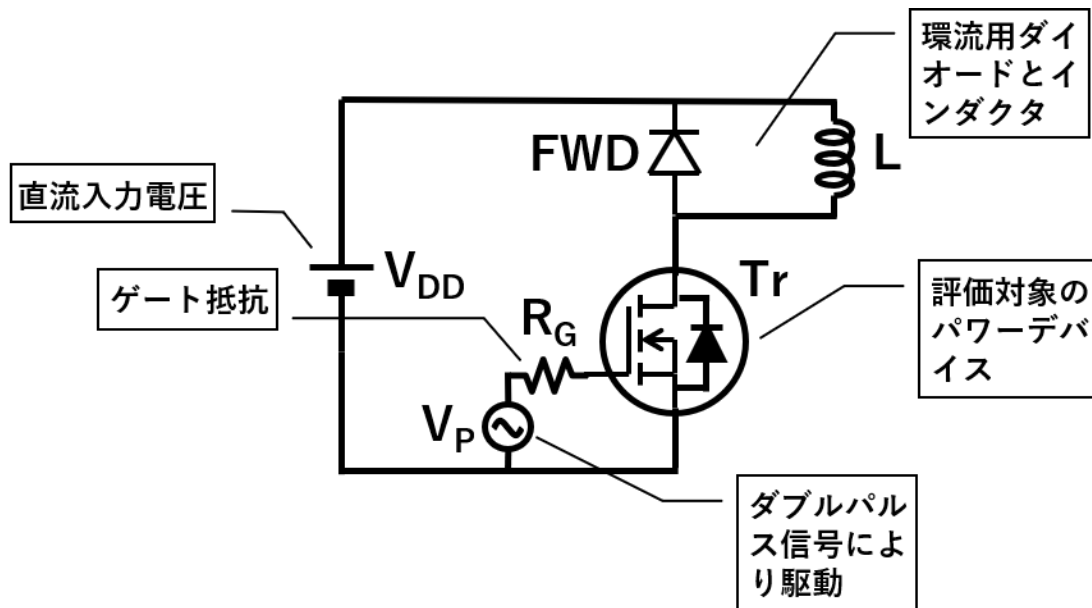


図 2-2 ダブルパルス回路

の傾きで線形的に増加していく。インダクタ電流が測定したい I_{c/I_d} になったときに、パワーデバイスをターンオフする。このときのパワーデバイスの端子間電圧や電流を観測すれば、測定したい I_{c/I_d} におけるターンオン特性が得られる。

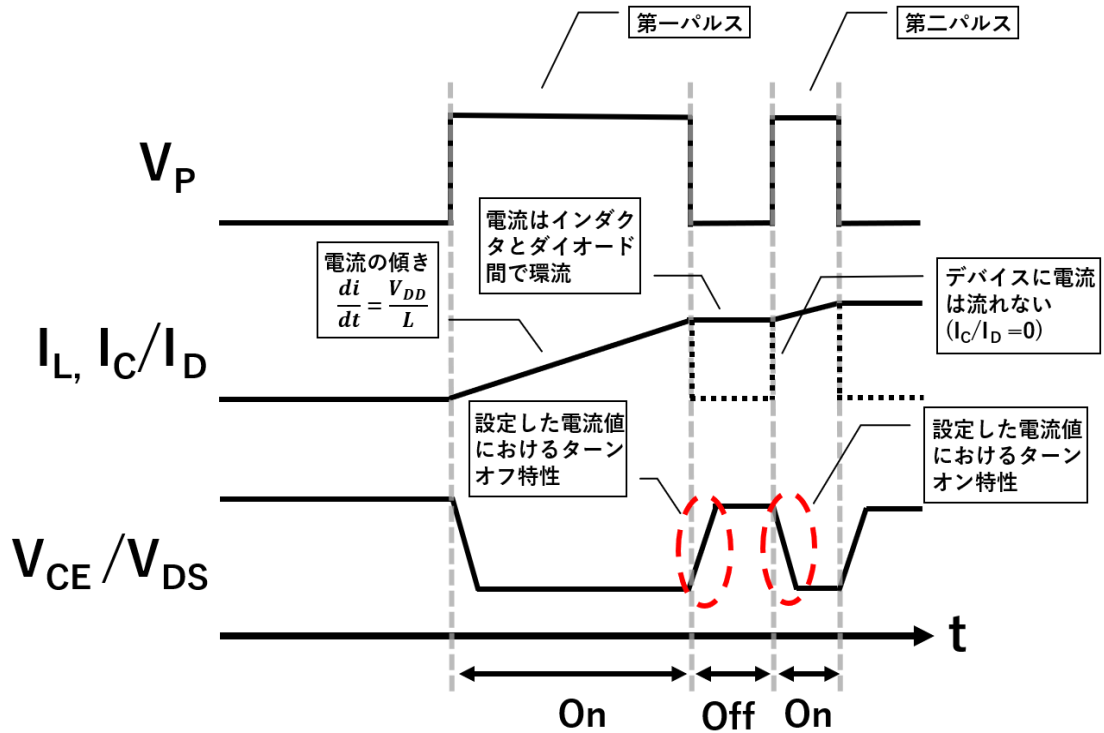


図 2-3 パワーデバイスの電圧と電流の変化

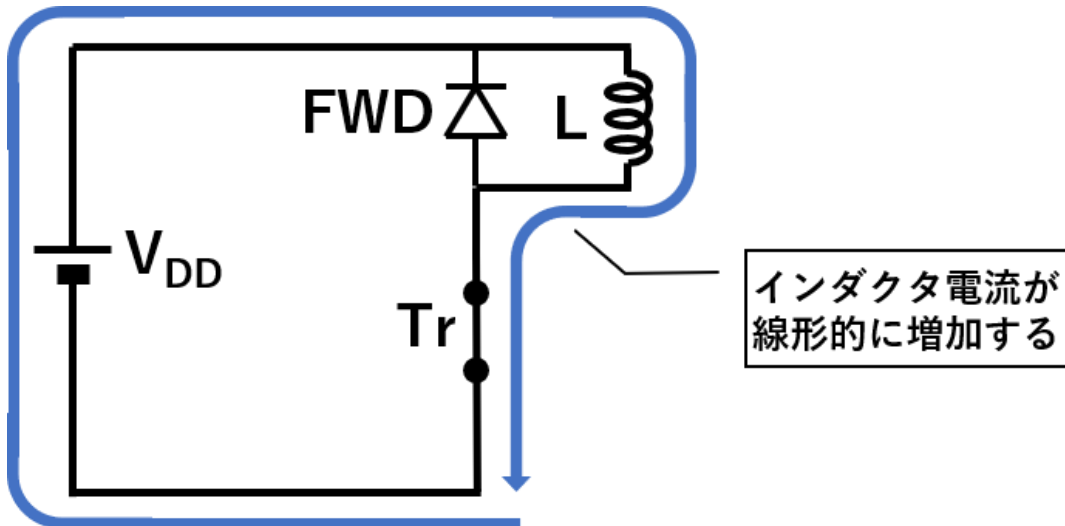


図 2-4 スイッチオン状態

2.2.2 ターンオフから二回目のターンオン時まで(第二パルス)

ターンオン特性の測定には、短いオフ期間を設けて、再度ターンオンする瞬間にパワーデバイスの端子間電圧や電流を観測する。オフ期間は図 2-5 の状態で、電流はインダ

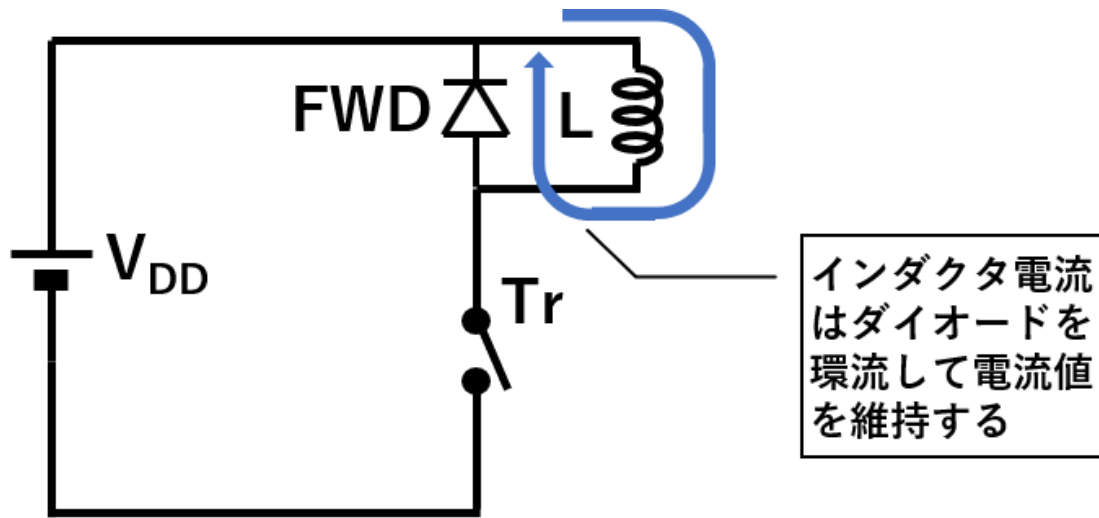


図 2-5 スイッチオフ状態

クタに蓄積され、ダイオードを介して環流する。このときの電流値は、オフ期間が短ければ、オフする前の値を維持しているため、ターンオフしたときの電流値とほぼ等しい。

パワーデバイスのスイッチング特性は、ターンオン・ターンオフするときの I_c/I_d に大きく依存するため、スイッチングを行う I_c/I_d を自由に設定できるようにする必要がある。測りたい I_c/I_d を設定しスイッチング特性を観測するのは、それを可能とするダブルパルス回路の大きなメリットである。

2.2.3 パワーデバイスのターンオン特性

パワーデバイスのターンオン特性は図 2-6 に示す[9]。まず、ゲート端子を電源電圧に接続し、ゲート電圧 V_g が上昇しはじめる。 V_g が t_1 で V_{th} を超えると、 I_c/I_d が流れはじめ、 $C_{GE/GS}$ が $t_1 \sim t_2$ 間に充電される。 $C_{GE/GS}$ が完全に充電され、 I_c/I_d が第一パルスで設定した電流値に到達すると、 $V_{CE/DS}$ が下がり始める。次に、 $t_2 \sim t_3$ の間は I_c/I_d が $C_{GC/GD}$ を充電する。 $V_{CE/DS}$ が下がり続けているため、 $C_{GC/GD}$ の充電時間は $C_{GE/GS}$ より長くなる。また、この間 V_g は一定に保たれていることから、プラトー領域とも呼ばれている。最後に、 $C_{GE/GS}$ と $C_{GC/GD}$ 充電が両方完了すると、 V_g が再び上昇し、 t_4 で電源電圧の値に到達する。

以上により、本研究の学習に用いる V_g の波形データは二回目のターンオン時より取得され、そのパワーデバイスのターンオン特性を学習対象としている。

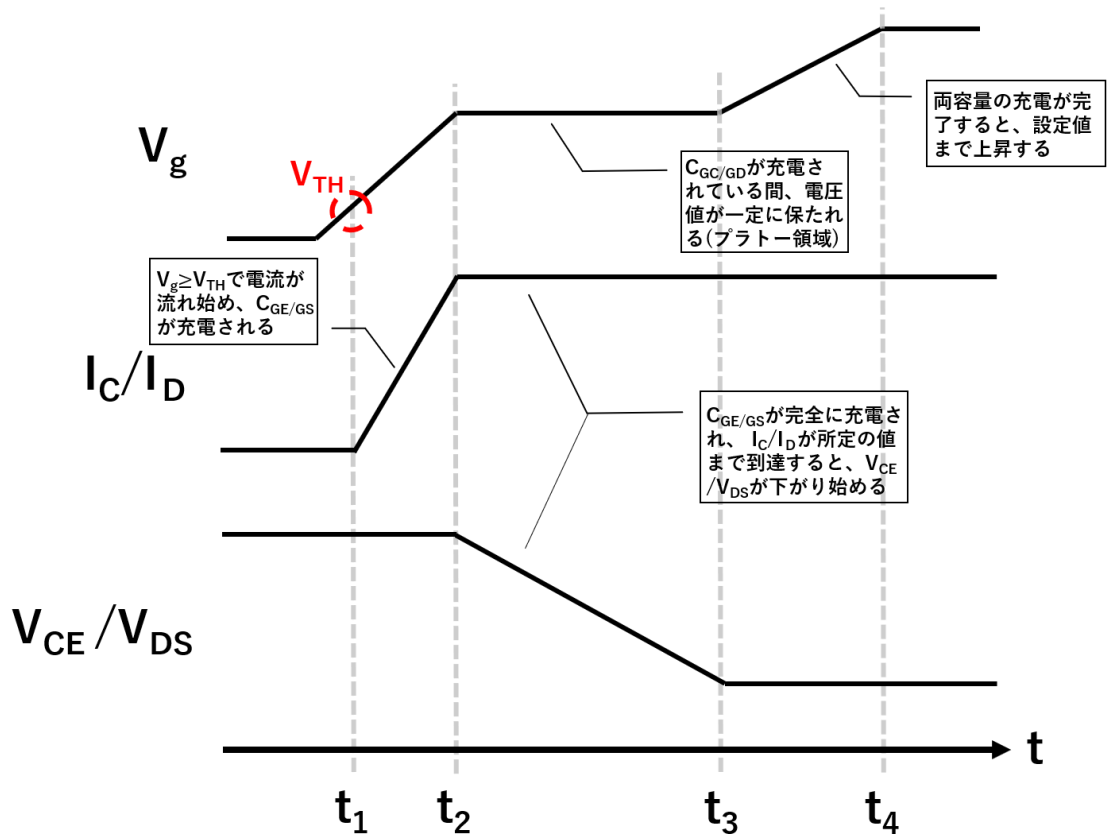


図 2-6 パワーデバイスのターンオン特性

第3章 機械学習とニューラルネットワーク

本章では、学習の流れおよび用いられた python 言語ベースの機械学習とニューラルネットワークライブラリ、および本研究で採用した分類と特徴抽出アルゴリズムについて説明する。

機械学習の問題は教師あり学習と教師なし学習に大きく二分できる。教師あり学習はさらに分類問題と回帰問題に分かれる。学習データセットが持つ特徴量とラベルの入出力関係について学習し、新しいデータに対し、離散的なラベル(カテゴリ、クラス)を予測する問題を分類、連続的なラベル(値、量)を予測する問題を回帰と定義されている。

一方、教師なし学習はラベルを持たず、クラスタリングとデータセットの変換に分かれる。クラスタリングはデータセット内に類似するデータ同士のグループを見つける問題である。データセットの変換は、元々のデータ表現を変換し、人間や他の機械学習アルゴリズムにとってより分かりやすいデータ表現にするアルゴリズムである。大量の特徴量で構成されている高次元なデータを少量の本質的な特徴量でそのデータ要約する方法を見つけることが一般的であるため、次元削減または特徴抽出とも言われている。また、特徴量の数を2か3次元に減らせば、データの可視化に用いられる。

3.1 学習の流れ/アルゴリズムの選択

機械学習やニューラルネットワークでの学習は、図 3-1 に示す手順で行われている。学習に使うデータを十分に収集し、必要ならばデータの前処理を行う。目的に応じて分類や回帰、クラスタリングに適した学習アルゴリズムを選択し、学習した結果をさまざまな基準で学習の出来具合を評価する。オープンライブラリでは、これらを行うに必要なアルゴリズムが一通り揃っている。

本研究でデバイスパラメータの推定に使用した三種類の学習フローを図 3-2 にまとめている。分類と特徴抽出アルゴリズムは `scikit-learn` ライブラリより `k`-最近傍法(`kNN`)、線形判別分析(`LDA`)、主成分分析(`PCA`)を採用した。`kNN` の仕組みは直観的に理解しやすく、特徴抽出を行った後のデータの評価が行いやすいという利点で採用に至った。`PCA` に関しては、特徴抽出アルゴリズムにおいて最もよく使われているため、これを用いた学習結果を基準とした。`LDA` は教師ありでありながら特徴抽出に使えることが特徴的であるために採用した。一方、`Keras` ライブラリより畳み込みニューラル

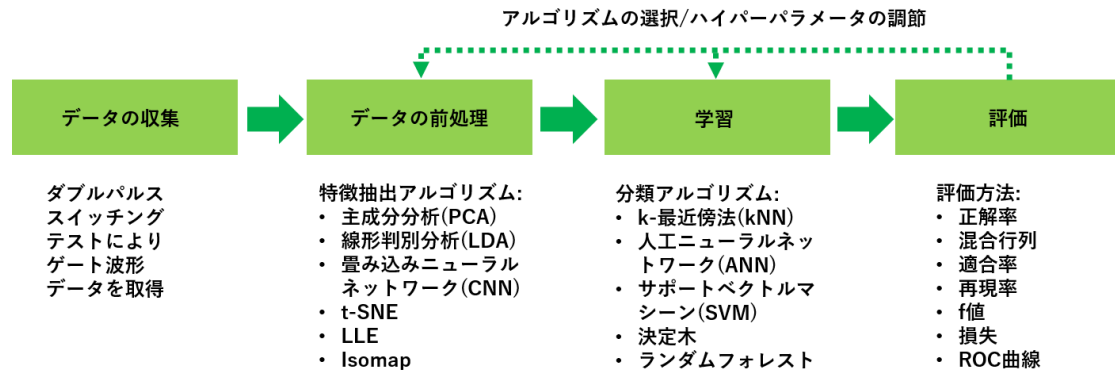


図 3-1 学習の流れおよびさまざまな学習アルゴリズム

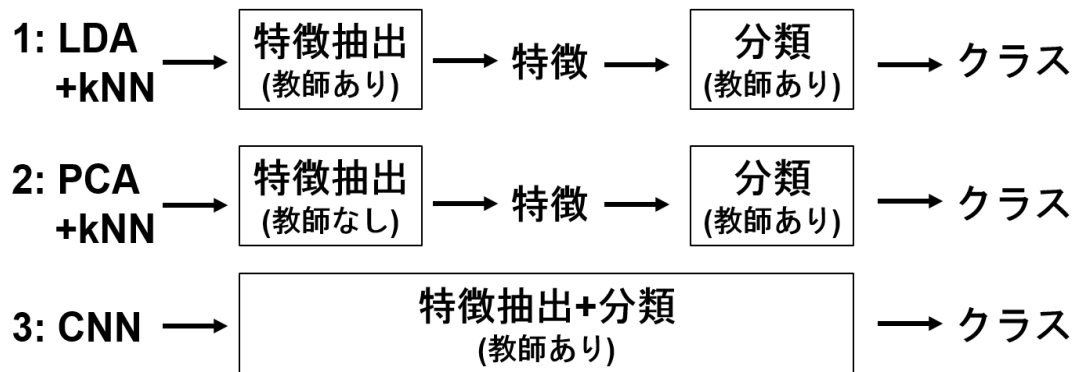


図 3-2 本研究で採用した学習フロー

ネットワーク(CNN)を採用した。理由として、近年 CNN は画像認識において注目されていることから、本研究の波形データについても画像を認識するように分類できるかどうかという観点によるものである。

3.2 学習データの特徴抽出アルゴリズム

3.2.1 主成分分析

主成分分析(PCA)は、学習データにもっとも接近する特徴空間を見つけ、そこにデータを射影する特徴抽出アルゴリズムである[10]。PCA は、まずデータセットの分散を最大限に維持する軸を見つける。次に、第一の軸と直交するもう一つの軸も見つけ、この軸は二番目に大きな分散を持つ。もっと次元の高いデータセットでは、さらにそれまでの二本の軸に直交する第三、第四あるいはそれ以上の軸も見つけてくる。第 i 番目の軸を定義する単位ベクトルを i 番目の主成分と呼ぶ。

図 3-3 に示す三次元の特徴空間に格納されているデータセットの例では、赤矢印が

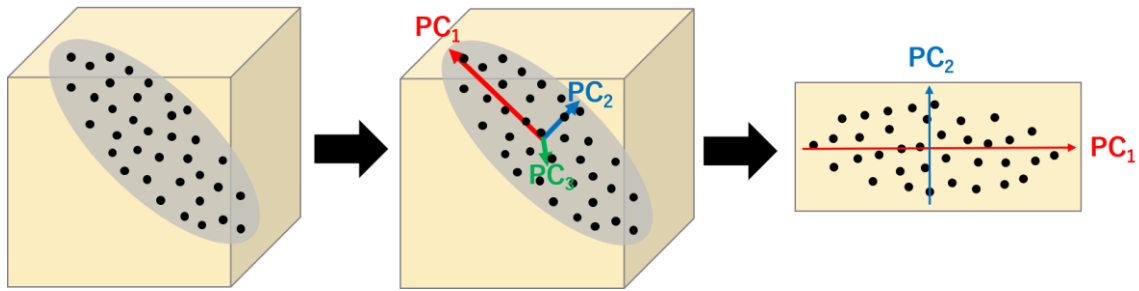


図 3-3 PCA を用いたデータ変換

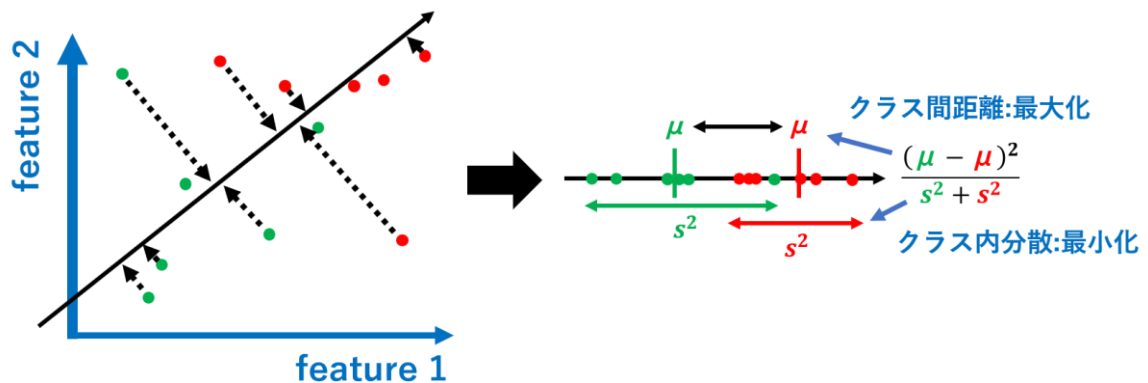


図 3-4 LDA を用いたデータ変換

第一主成分に当たり、赤矢印に直交する青矢印を第二主成分、第三主成分の緑矢印まで軸を定義することができる。第三主成分はわずかな分散しか持たないことから、データセットを第一・第二主成分のみで構成された二次元の新たな特徴空間(平面)に射影しても、データセットが持つ元々情報量を最大限に維持した次元削減が行うことができる。

3.2.2 線形判別分析

線形判別分析(LDA)は元々分類アルゴリズムであるが、次元削減テクニックとして優れているため、教師ありの特徴抽出アルゴリズムとして扱うこともできる[11]。LDAは異なるラベルのデータセットをもっとも特徴的に分けられるような軸を見つけ、それらの軸を使ってデータを射影する新たな特徴空間を定義する。利点として、射影によってラベルができる限り遠く引き離されることが挙げられる。

図 3-4 に示すデータセットを二つのクラスを分類する例では、赤ラベルと緑ラベルを持つデータのクラス間距離を最大にし、クラス内分散を最小にするように新たな軸を定義することで、二つのクラスをもっとも離れるようにデータセットを射影することができる。

3.2.3 畳み込みニューラルネットワーク

通常のニューラルネットワークの学習方法は空間的特徴、つまりデータの隣り合う特徴量同士の関連性を捉えることができない。畳み込みニューラルネットワーク(CNN)はこの欠点を補うためのネットワークである。CNNは畳み込みを行う畳み込み層とサイズ圧縮を行うプーリング層が交互に重なり、最後に通常のニューラルネットワークに接続するように構成されたネットワークである。

図 3-5 のような手書き数字の例を挙げると、畳み込み層のニューロンは、入力画像の一定領域にあるピクセルと接続するようにする[12]。この一定領域の情報を一つのニューロンに集約して接続する処理を畳み込みと呼ぶ。畳み込み領域を1ピクセルずつずらして処理していくと、複数の特徴マップが作成される。次に、作成された特徴マップを扱いやすくするためにサイズを圧縮する。圧縮には特徴マップの一定領域内の最大値を取る(最大値プーリング)や平均を取る(平均値プーリング)という主な二つの手法がある。畳み込みとプーリングを必要な回数だけ繰り返し、新たな特徴量が抽出される。本研究はゲート波形の時系列データを扱っているため、図 3-6 と図 3-7 のように一次元の畳み込み処理を行う。

3.3 学習データの分類アルゴリズム

3.3.1 k-最近傍法

k-最近傍法アルゴリズム(kNN)は、最も単純な学習アルゴリズムであるといわれている[10]。訓練データセットを特徴空間に格納するだけで構築できる。新しいデータに対する予測は、訓練データセットの中から最も近い点「最近傍点」を見つけることで行われる。kの値を設定することで、任意個の最近傍点で投票を行い、ラベルを決める。図 3-8 の特徴空間で例を挙げると、k=3 の場合は赤三角と、k=5 の場合は青四角のように、kの値によって異なる予測結果を得ることが考えられる。

3.3.2 人工ニューラルネットワーク

人工ニューラルネットワークネットワークは、隣接する層で相互接続されたニューロンで構成されている[13]。より高い正確さが得られるようにニューロン同士の接続方法を探索し、そのモデルを学習することを目的としている。図 3-9 に示す手書き数字の認識の例を再び挙げると、画像の各ピクセルが持つ白黒の度合いを特徴量とし、一列に広

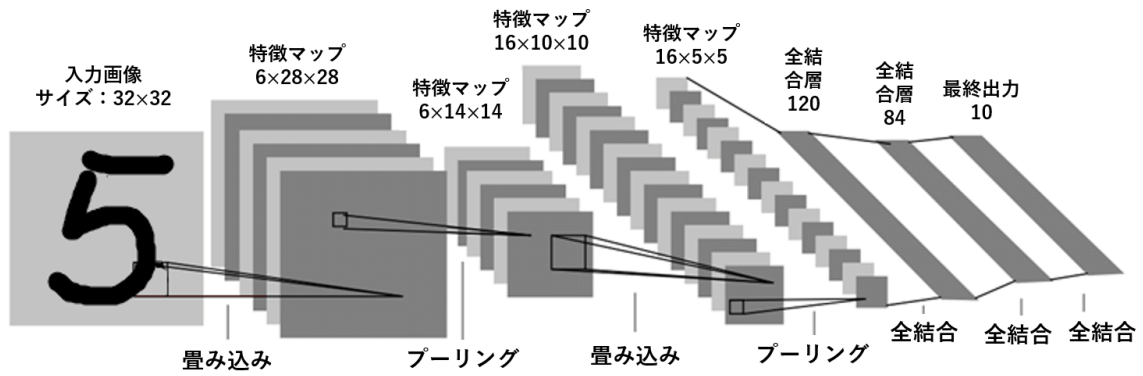


図 3-5 二次元畳み込みニューラルネットワークの構造[12]

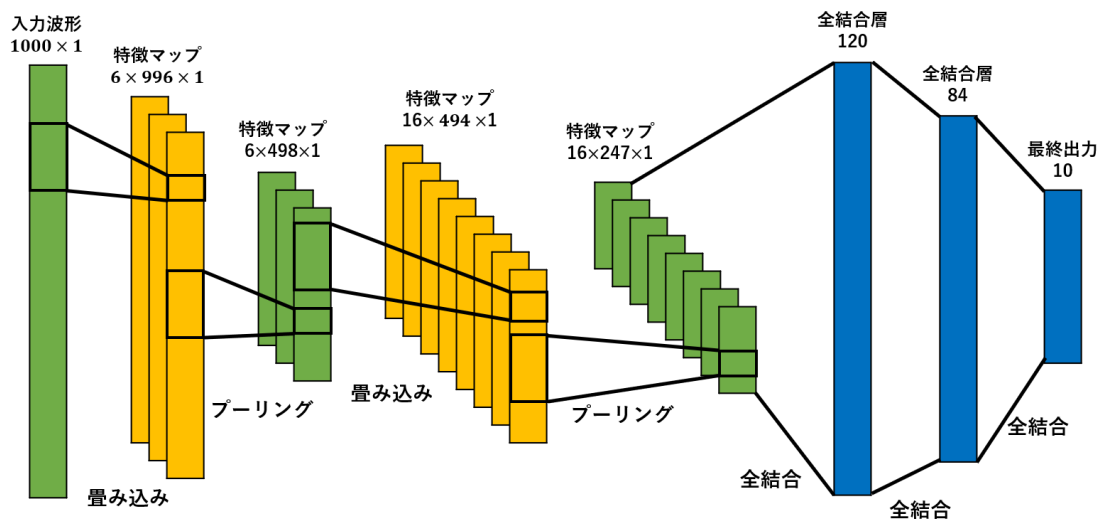


図 3-6 一次元畳み込みニューラルネットワークの構造

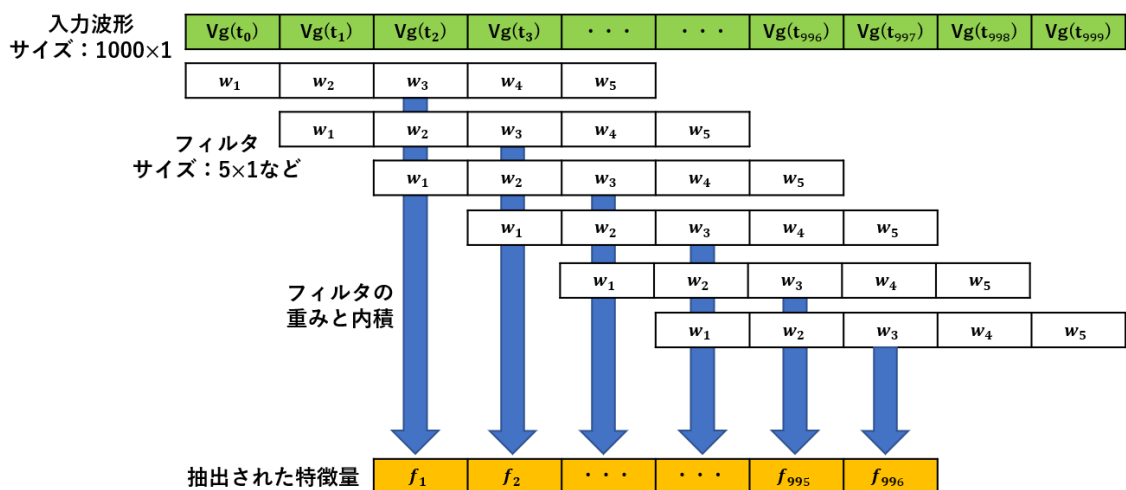


図 3-7 時系列(一次元)の畳み込み

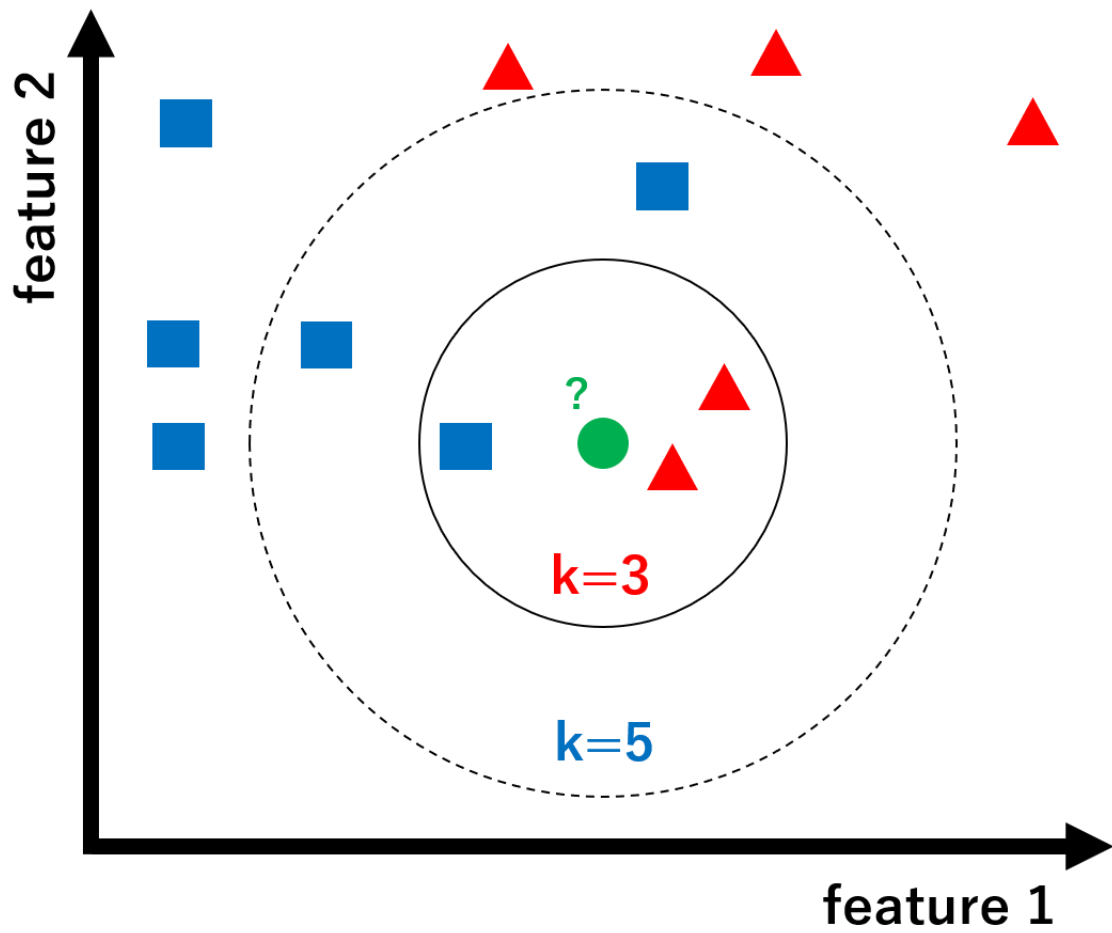


図 3-8 kNN アルゴリズムの予測

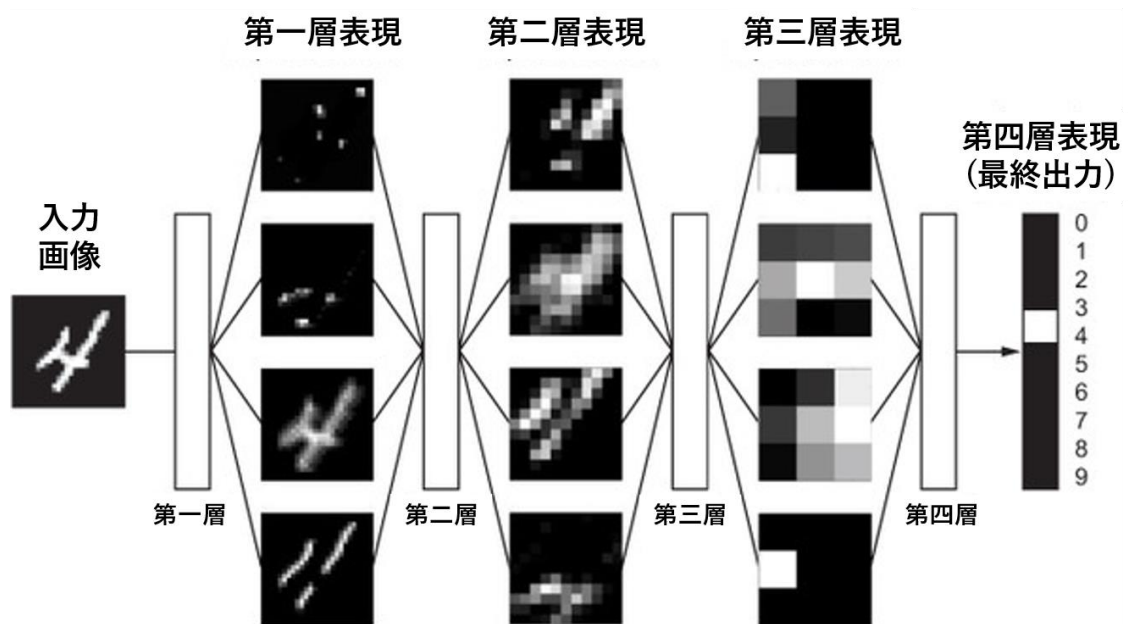


図 3-9 数字を分類するためのニューラルネットワーク[15]

げてネットワークの入力層に入力され、最終的にラベル 0~9 のいずれかから結果が出力される。この例では、ネットワークは数字の画像を新たな表現に変換し、それぞれの表現は元の画像から徐々に異なるものに変化しながら、最終的な結果に関する情報を少しずつ増やしていくように学習する。

3.4 オープンソースライブラリ

Python に実装されているライブラリでは、データを 2 次元配列または行列に格納している。データ行列 \mathbf{X} のサイズはサンプルの数(`n_samples`)とそれぞれのサンプルが持つ特徴量の数(`n_features`)の積となっており、教師あり学習の場合、ラベルベクトル \mathbf{y} は `n_samples` と同じサイズを持つ。

3.4.1 scikit-learn

scikit-learn[14]の API は、すぐに一般的な機械学習アルゴリズムにアクセスできるように設計された、大学でも産業界でも広く使われている、機械学習分野において重要なライブラリである。誰でも簡単に機械学習を行えることを可能とした scikit-learn の主な特徴は以下のとおりである[10]。

- 一貫性: すべてのオブジェクトが推定、変換、予測の順に首尾一貫した単純なインターフェイスを持つ。
 - 推定器: データセットに基づいてパラメータを推定できるオブジェクトは推定器と呼ばれる。推定自体は `fit()` メソッドが呼び出されたときに行われており、引数はデータセット(教師あり学習アルゴリズムの場合、データセットとラベル)である。
 - 変換器: 一部の推定器は、データセットを変換することもできる。このようなものを変換器と呼ぶ。変換は変換対象のデータセットを引数とする `transform()` メソッドで行われる。戻り値は、変換後のデータセットである。
 - 予測器: 一部の推定器は、データセットを与えられると、予測することができる。このようなものを予測器と呼ぶ。予測器には、新しいインスタンスを格納するデータセットを引数とし、対応する予測のデータセットを返す `predict()` というメソッドがある。予測器には、テストセット(教師あり学習アルゴリズムの場合はテストセットと対応するラベル)を引数として予測の正確さを測定する `score()` メソッドもある。
- インспекション: 推定器のハイパーパラメータは、すべて公開インスタンス変数を通じて直接アクセスでき、推定器の学習後のパラメータは、すべてアンダースコアをサフィックスとする公開インスタンス変数を介してアクセスできる。

- クラス増加の抑制:データセットは、自家製のクラスではなく、**Numpy** 配列または **SciPy** 疎行列で表現される。ハイパーパラメータは、**Python** の通常の文字列または数値である。
- 合成:既存のコードはできる限り再利用できるようになっている。例えば、一連の無作為な変換器のあとに最後に推定器が続く推定器のパイプラインを簡単に作れる。
- 妥当なデフォルト値:**scikit-learn** は、ほとんどのハイパーパラメータに妥当なデフォルト値を設定しており、動作する基本システムをすばやく作れるようになっている。

3.4.2 Keras

Keras[15]は、モジュール式で、最小限で容易に拡張可能なアーキテクチャでほぼあらゆる種類のニューラルネットワークモデルを定義して訓練するためのフレームワークである。**Keras** の利点を以下に示す[13]。

- モジュール性:**Keras** のモジュールはレゴブロックのように組み合わせてニューラルネットワークを構築できる。すなわち、事前に定義されたさまざまな種類のニューラルネットワーク層、損失関数、最適化アルゴリズム、初期化方法、活性化関数、および正則化方法を組み合わせてネットワークを定義できる。
- ミニマリズム:各モジュールは短く自己記述的になっている。
- 拡張性:ユーザーは必要に応じて **Keras** を拡張することができる。
- CPU でも GPU でも同じコードをシームレスに実行できる
- ニューラルネットワークモデルのプロトタイプを簡単にすばやく作成できるユーザーフレンドリな **API** を備えている。
- 畳み込みネットワーク、リカレントネットワーク、およびそれらの組み合わせを組み込みでサポートしている。
- 複数入力/複数出力モデル、層の共有、モデルの共有など、任意のネットワークアーキテクチャをサポートしている。つまり、ほぼすべてのディープラーニングモデルの構築に適している。

第4章 パワーデバイスのパラメータ推定

本章では、デバイスパラメータを推定するために提案された先行研究を紹介し、本研究の提案手法を述べる。もっとも原始的な方法として、劣化しやすい箇所にさまざまなセンサを直接取り付け、その後パワーデバイスのデータシートを参照しながら判定することが行われていた。しかし、より正確かつリアルタイムのモニタリングを実現するために、近年は図 4-1 のように主回路から、あるいは図 4-2 のように制御回路から取得した電圧・電流データをさまざまなデータ解析手法で間接的に推定することが一般的な傾向である。

4.1 先行研究

4.1.1 IGBT の劣化判定

IGBT の劣化にもっとも大きく反映にされるデバイスパラメータはターンオン時のコレクタ・エミッタ間電圧($V_{CE(ON)}$)である。Xiong らは $V_{CE(ON)}$ が初期値より 15~20%ほど増加していれば、その IGBT は寿命に達するしきい値であることを結論付けた[16]。しかし、Sutrisno らのパワーサイクル試験結果によると、IGBT の寿命と $V_{CE(ON)}$ の増加関係には個体差が存在することを示した[17]。図 4-3 のように、IGBT の寿命の 94%や 80%に時点で $V_{CE(ON)}$ の増加しきい値に達した場合もあれば、寿命が越えていても $V_{CE(ON)}$ の増加はしきい値に達していない場合もあることから、劣化の見逃しを招く恐れがあることを示唆した。

一方、Patil らは $V_{CE(ON)}$ および $I_{CE(ON)}$ を取得し、マハラノビス距離やカイ二乗分布といった統計学的手法に適用した[4]。まず、図 4-4 のように IGBT から初期のデータセットを作り、劣化していない健康領域とする。そこから標準偏差の三倍(3σ)を超えるマハラノビス距離を持つデータ、あるいはカイ二乗分布の上側 2.5%に入るデータを持つ IGBT を劣化と判定した。しかし、この健康領域は一定のジャンクション温度で取得した $V_{CE(ON)}$ と $I_{CE(ON)}$ によって定められている。パワーデバイスは動作中に温度が変化し続けているため、デバイスの温度特性が考慮されていない健康領域での判定は劣化の誤認をする可能性が高いことを図 4-5 で示されている[17]。

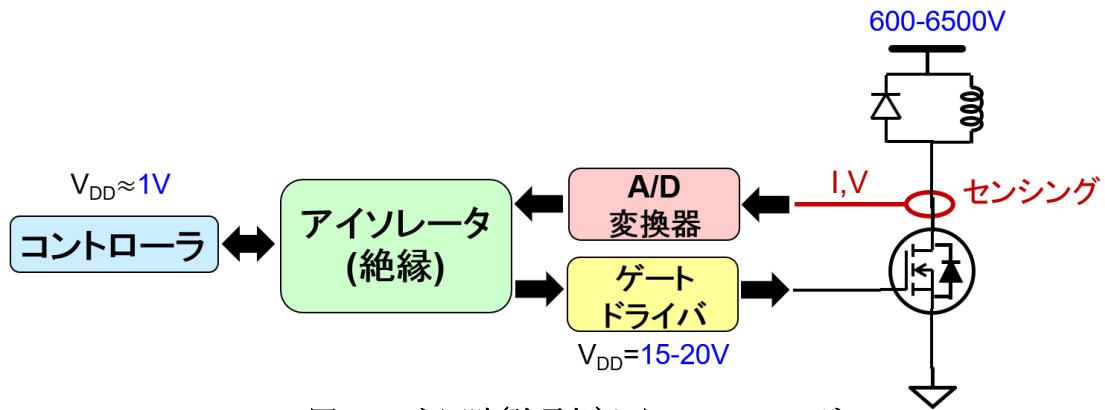


図 4-1 主回路(強電部)からのセンシング

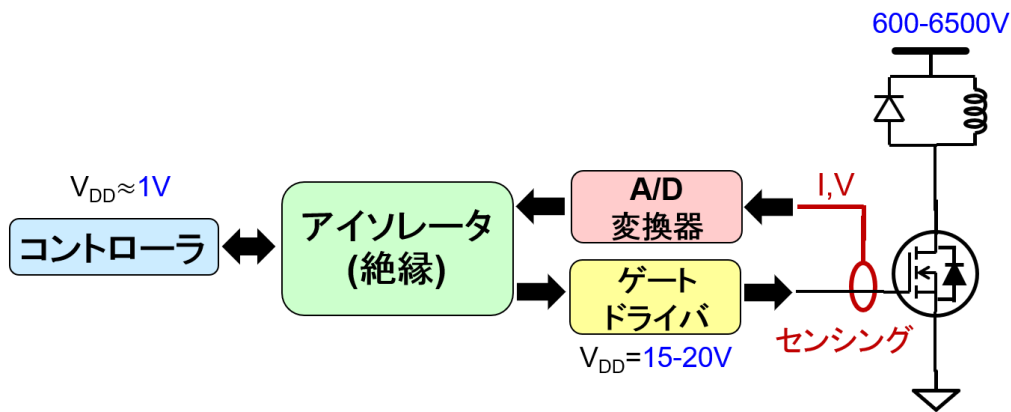


図 4-2 制御回路(弱電部)からのセンシング

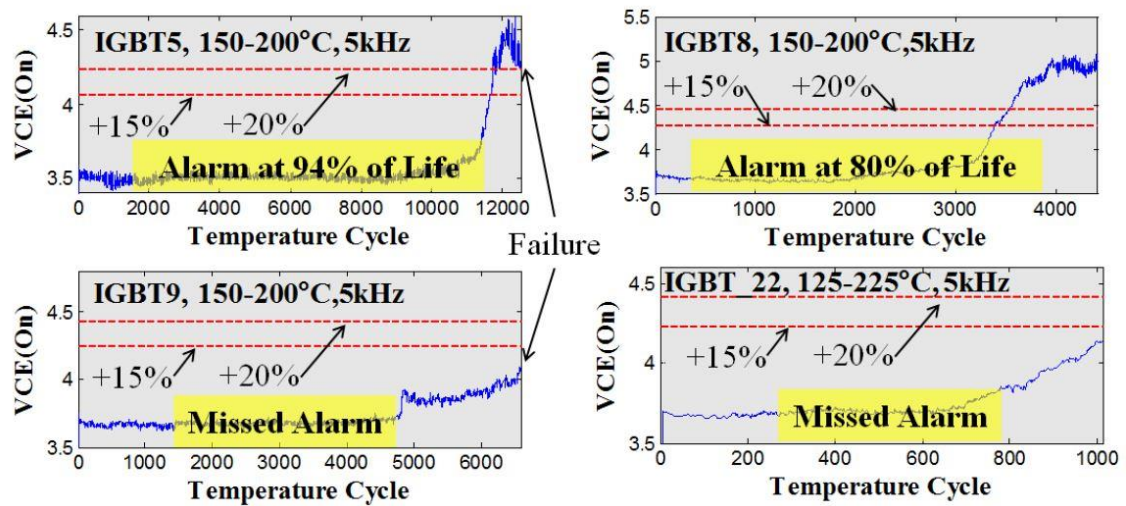
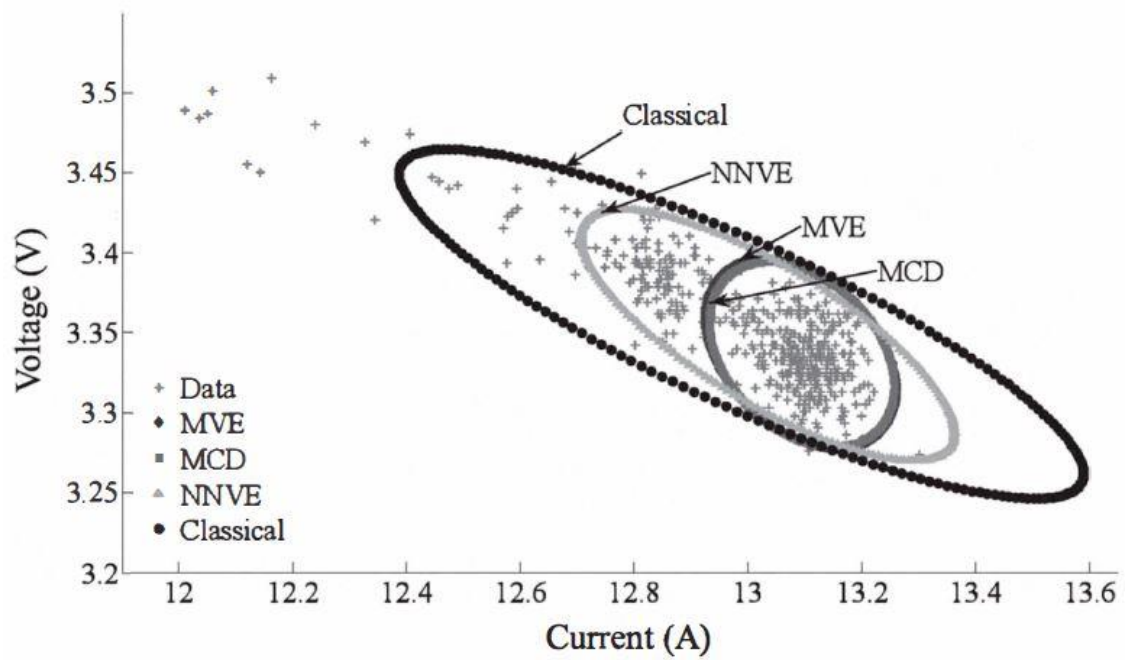
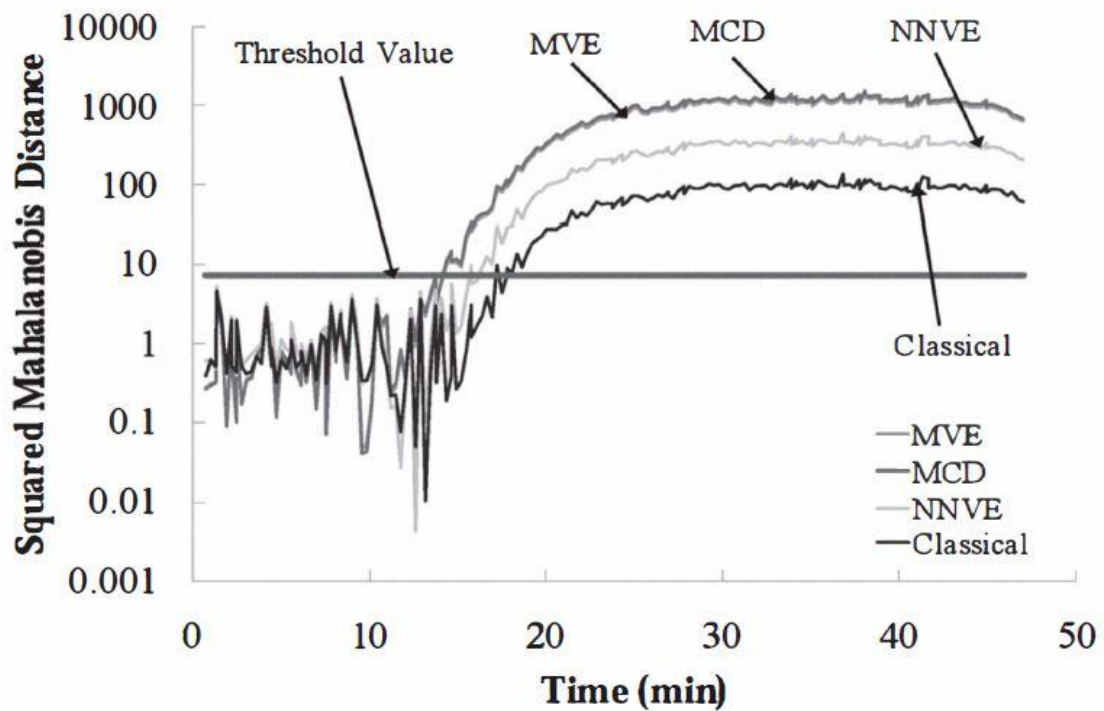


図 4-3 $V_{CE(ON)}$ の増加を用いた劣化判定の失敗例[17]

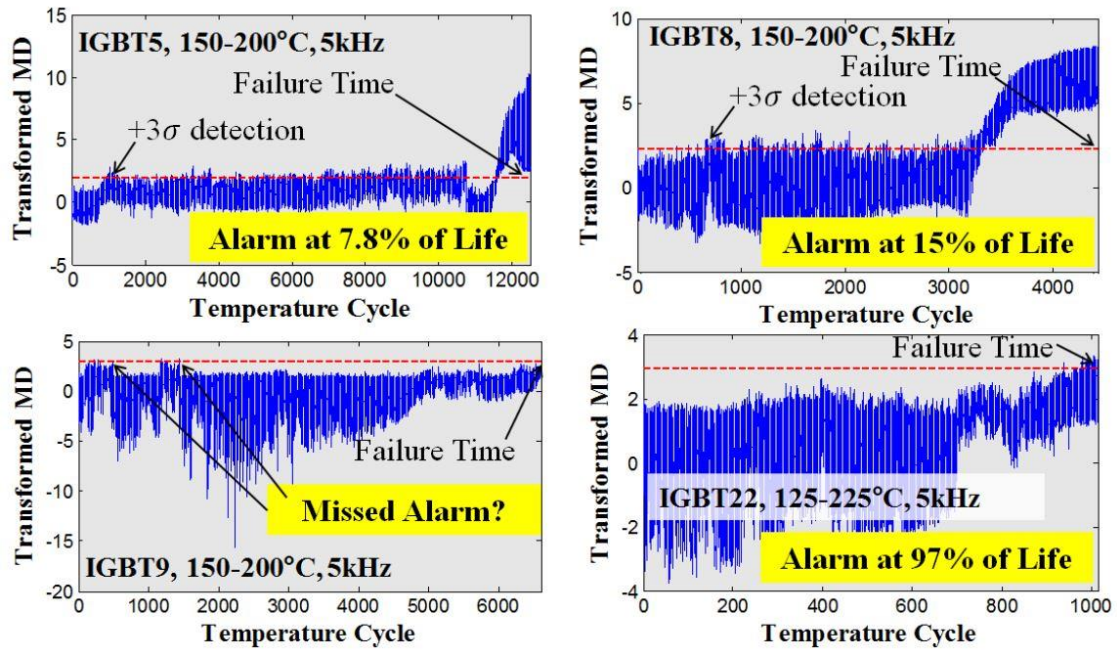


(a) $V_{CE(ON)}$ と $I_{CE(ON)}$ で定めたパワーデバイスの健康領域

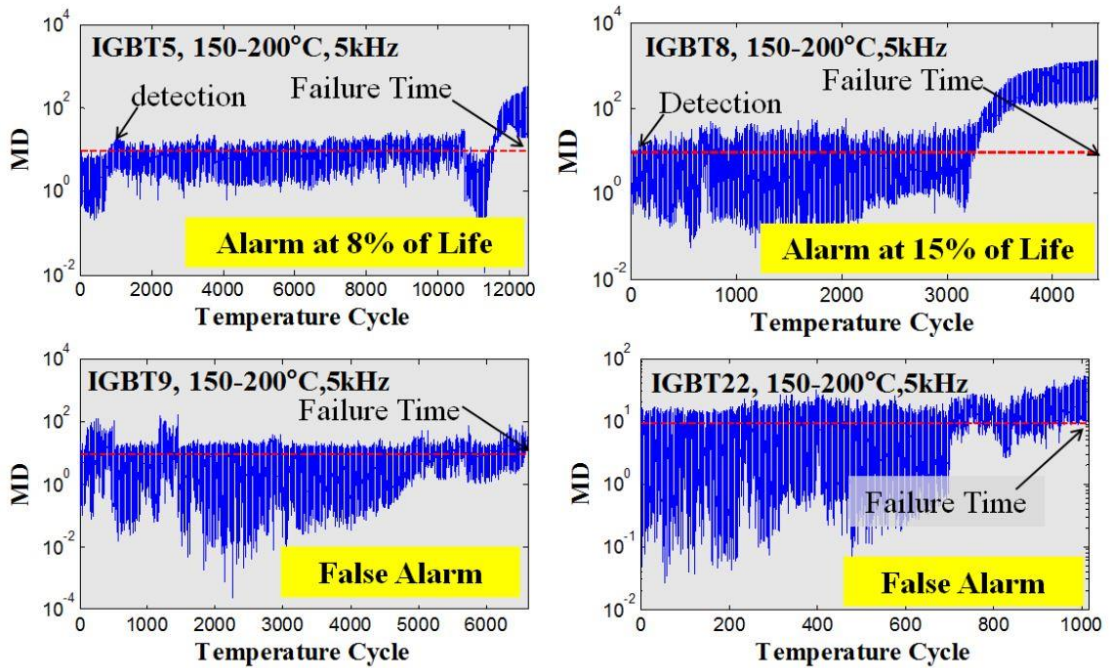


(b) マハラノビス距離の計算

図 4-4 統計学的手法を用いた劣化判定[4]



(a) マハラノビス距離の 3σ を劣化しきい値とした判定



(b) カイ二乗分布の上側 2.5%を劣化しきい値とした判定

図 4-5 図 4-4 で用いた劣化判定の失敗例[17]

4.1.2 ジャンクション温度の推定

理想的に主回路から絶縁されたゲート端子の電圧値と電流値は、パワーデバイスの寄生容量(C_{gc}/C_{gd} , C_{ge}/C_{gs})により、主回路の動作条件に伴って変動する。この性質を用いたジャンクション温度の推定手法が提案されている。Bakerらはターンオン時のゲート抵抗における電圧降下をセンシング箇所とし、図 4-6 に示す測定回路をゲートドライバに組み込み、電圧波形のピーク値を検出する[5]。図 4-7 のように、ジャンクション温度によって電圧波形が少し変化するため、そのピーク値の違いから温度を推定する手法である。しかし、波形の一点のみをモニタリングすることは、ノイズや負荷変動などによる影響を受けやすいことを意味する。また、ピーク値の微小な変化を見分けるには 12-14bit のような高性能の A/D 変換器が必要であることが問題点として挙げられる。

4.2 提案手法

以上のより、パワーデバイスのスイッチング動作は複雑であるために、デバイスパラメータの推定方法やその解釈に注意を払う必要がある。そうでなければ、現実的な結果が得られないことや、誤認や見逃しによる部品の浪費と予定外の故障など、本来の目標であるパワーデバイスの信頼性向上が達成できなくなる。

したがって、本研究はゲート端子をセンシング箇所とし、ターンオン時から過渡現象が終了するまでの電圧波形を取得する。波形の各タイミングにおける電圧値を特徴量とした一つの高次元データを、機械学習やニューラルネットワークアルゴリズムで学習する。ターンオン時のある特定の一点のみで判定するのではなく、波形全体の時系列的な挙動で判定すれば、よりロバストなモニタリングが可能となると考えられる。

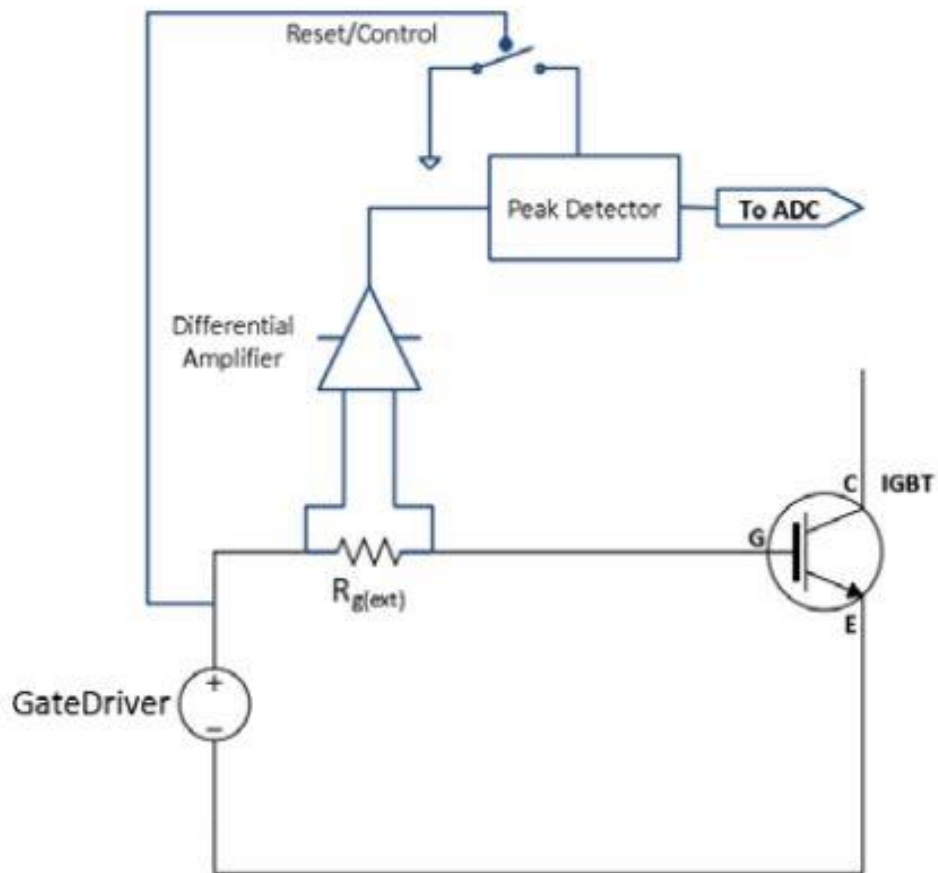


図 4-6 ゲート抵抗における電圧降下ピーク値の検出回路[5]

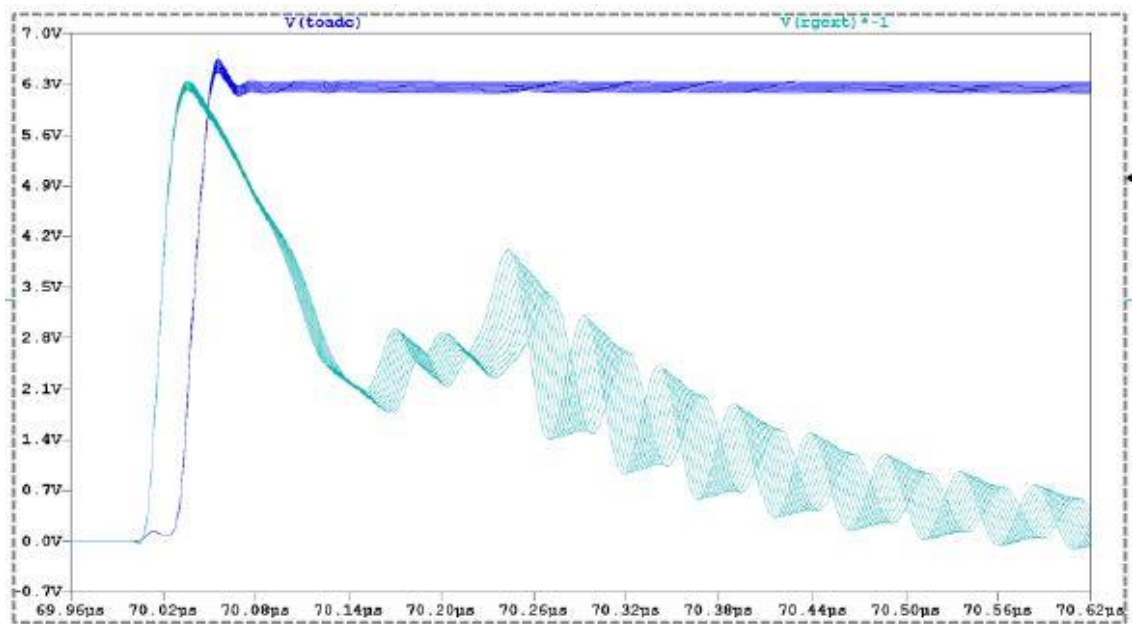


図 4-7 電圧波形および検出回路の出力[5]

第5章 学習データの取得

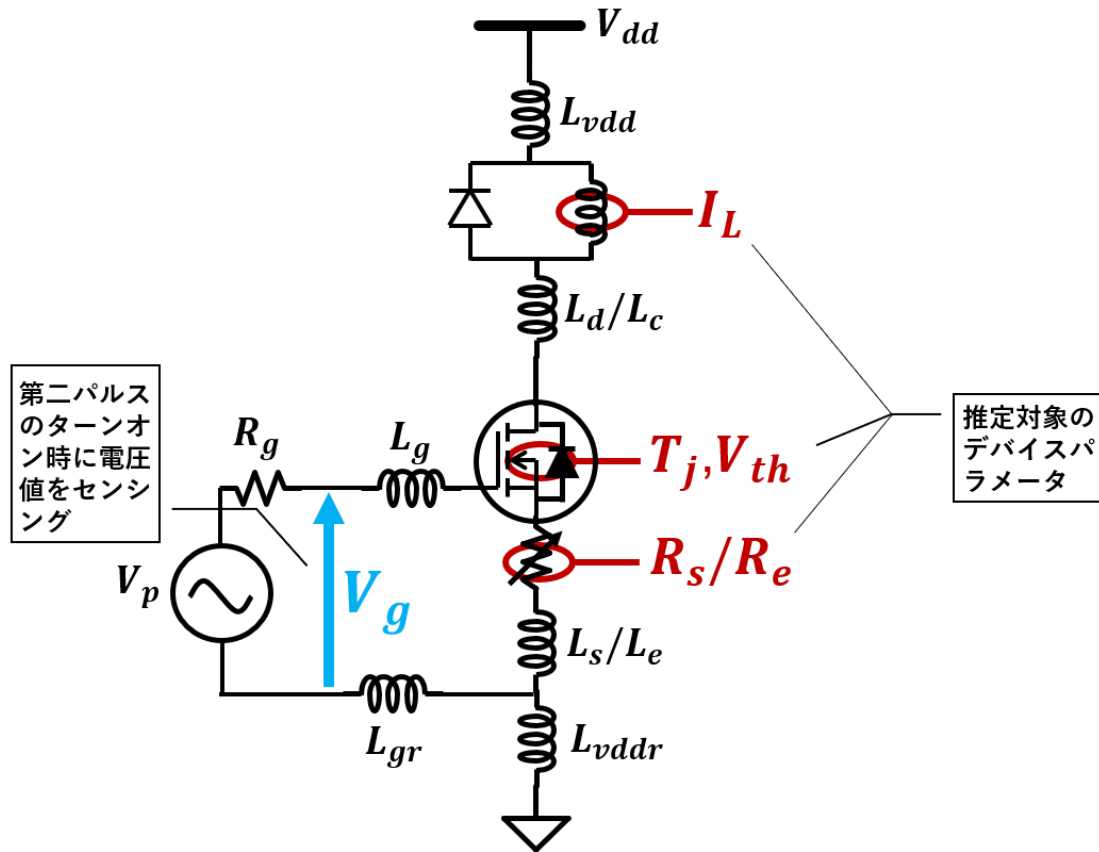


図 5-1 提案手法のセンシング箇所と推定パラメータ

図 5-1 に提案手法の波形データを取得するダブルパルス回路を示す。パワーデバイスの V_{th} や T_j 、およびボンディングワイヤにおける R_s/R_e 、負荷に流れる I_L がさまざまな値に変化する中で取得した波形データでその有効性を検証する。さらに、将来的にゲートドライバ IC 内で実装することを目標に、IC 内の A/D 変換器のサンプリング間隔や分解能を考慮した波形データを生成し、十分な正解率を得るために必要な A/D 変換器性能を評価する。

5.1 波形データの取得手順

5.1.1 LTspice シミュレーション

シミュレーションで取得した波形の設定条件を表 5-1 に示す。ただし、ダブルパルス回路の寄生インダクタンスを考慮し、6 箇所に付け加えている。

表 5-1 シミュレーション波形の取得条件

回路素子・動作条件	型番・値
パワーデバイス(SiC MOSFET)	C2M0025120D
環流ダイオード	C4D20120A
環流インダクタ	DKIH-3252-40D5
寄生インダクタンス	
L_{vdd}	6nH
L_d	3nH
L_s	6nH
L_{vddr}	6nH
L_g	6nH
L_{gr}	3nH
固定のパラメータ	
V_{dd}	300V
R_g	6.7 Ω
V_p	ON:20V OFF:0V
センシング時間/サンプリング間隔	1 μ s/1ns
変動するパラメータ	
I_L	1~100A
T_j	25~150 $^{\circ}$ C
V_{th}	0~5V
R_s	0~10m Ω

SiC MOSFET と環流ダイオード、環流インダクタはメーカーから提供された spice モデルを使用した。推定対象である四つのパラメータについては、 R_s の値を可変にし、 I_L は第一パルスの ON 時間で調節した。SiC MOSFET の spice モデルでは、 T_j と V_{th} の設定が可能であったため、これを利用した。また、シミュレーションの動作を確認するための回路を図 5-2 に示す。

5.1.2 高電圧測定

高電圧測定で取得した波形の設定条件を表 5-2 に示す。今回の測定は一個の IGBT モジュールに対して行ったため、 I_L と R_e のみ推定対象にした。 I_L は同じく第一パルスの ON 時間で調節し、 R_e については図 5-3 に示す取り付けパーツを入れ替えてそれぞれ 100 回ずつ測定した。取り付けパーツに 10m Ω の抵抗が 0 個から 5 個まで並列接続

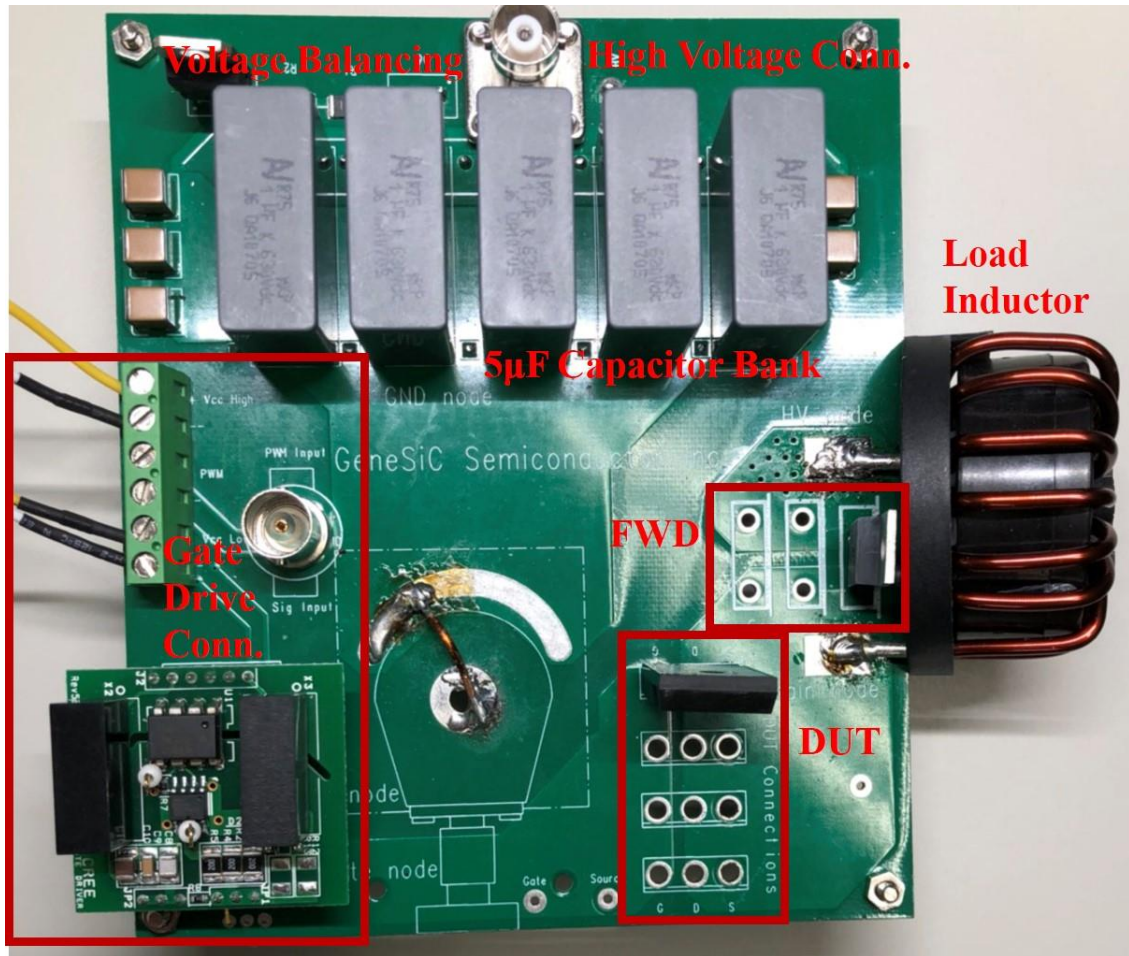


図 5-2 シミュレーション動作確認用測定回路

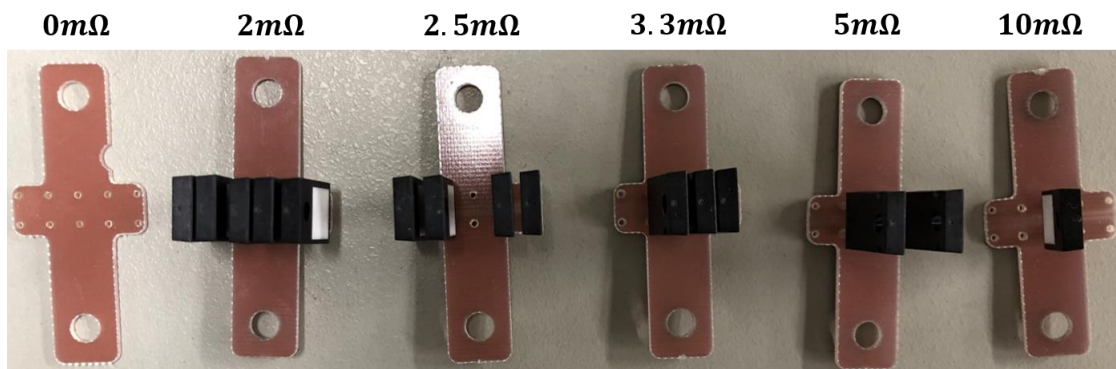


図 5-3 高電圧測定用可変 R_e の取り付けパーツ

されているため、 $R_e = [0, 2, 2.5, 3.3, 5, 10] \text{ m}\Omega$ のときの波形を取得した。

表 5-2 実測波形の取得条件

回路素子・動作条件	型番・値
パワーデバイス(IGBT モジュール)	2MBI100TA-060-50
環流ダイオード	モジュール内に付属
環流インダクタ	1.5mH
固定のパラメータ	
V_{dd}	300V
V_p	ON:15V OFF:0V
センシング時間/サンプリング間隔	3 μ s/1ns
変動するパラメータ	
I_L	2~100A
Re	0~10m Ω

5.2 取得した波形データの見たい目

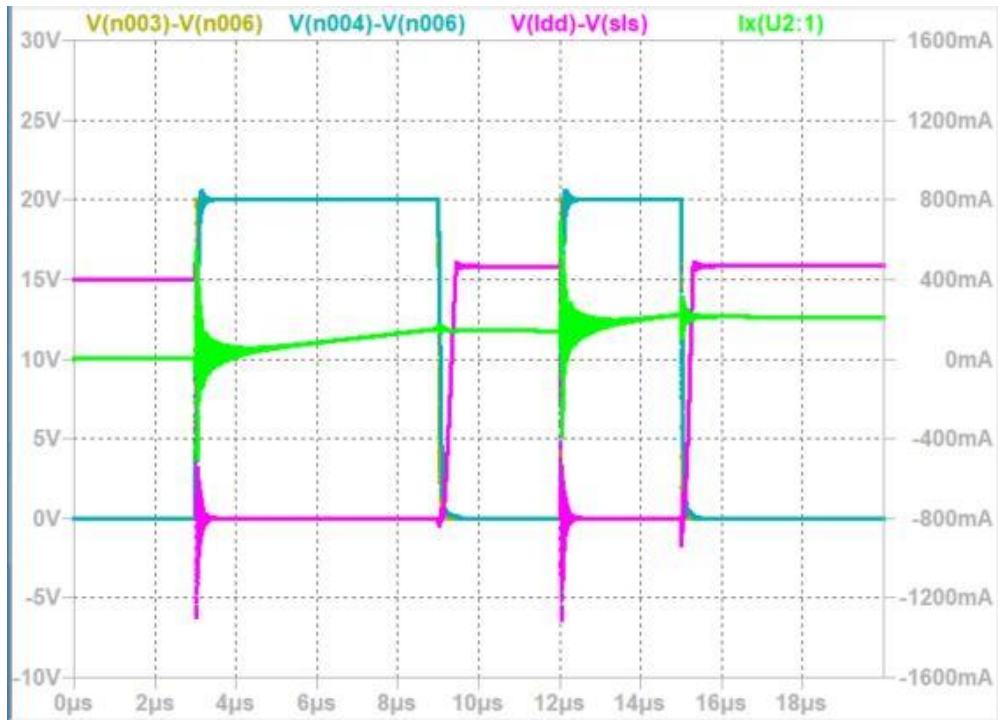
5.2.1 シミュレーション波形の確認

図 5-4 にシミュレーションで取得した入力信号(黄)および V_g (水)、 V_D (マゼンタ)、 I_D (緑)波形を示す。これにより、シミュレーション波形は実測とほぼ同等の動作を示していることを確認した。

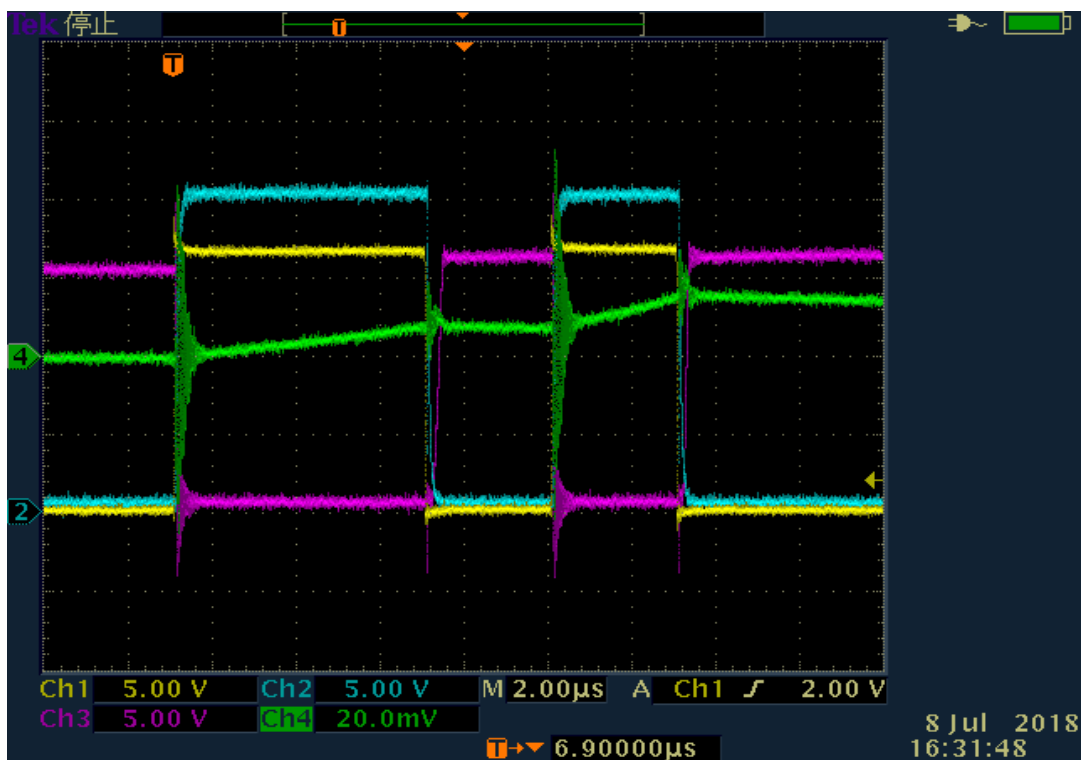
次に、図 5-5 にシミュレーションおよび高電圧測定で取得した V_g と V_D/V_C 、 I_D/I_C 波形を示す。他のデバイスパラメータを一定にし、 $I_D/I_C=10A$ と $20A$ のときの、第一パルスのターンオフ直前から第二パルスのターンオン直後までの波形を比較している。シミュレーションでは SiC MOSFET、実測では IGBT モジュールを使用しているため、ターンオン・ターンオフ特性は異なるものの、図 2-3 のようなダブルパルステストにおける一般的なパワーデバイスの動作を示しており、かつパラメータや動作条件によってターンオン・ターンオフ時の波形が変化することが確認できる。

5.2.1 高電圧測定波形の確認

図 5-6 に高電圧測定で取得したゲート電圧波形の全パターンを示す。 I_L と Re による波形の影響を見やすくするため、片方だけ変化したときに重ねられた図を図 5-7 および図 5-8 に示す。図 5-3 の取り付けパーツによる長い配線のせいで、寄生インダクタ

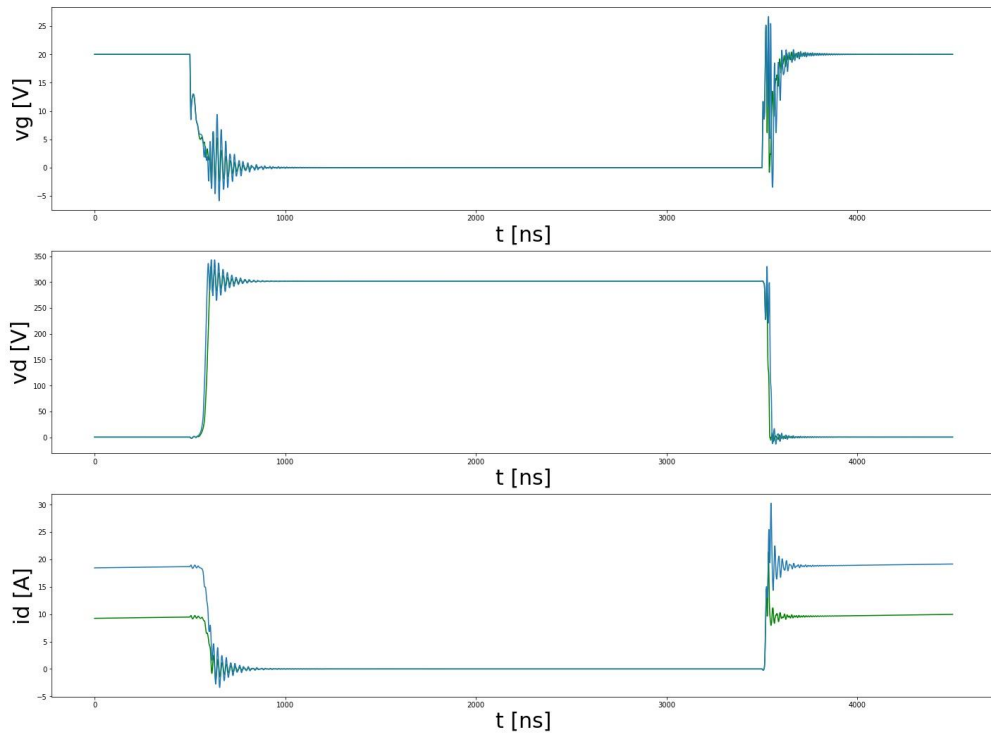


(a) シミュレーションで取得した波形

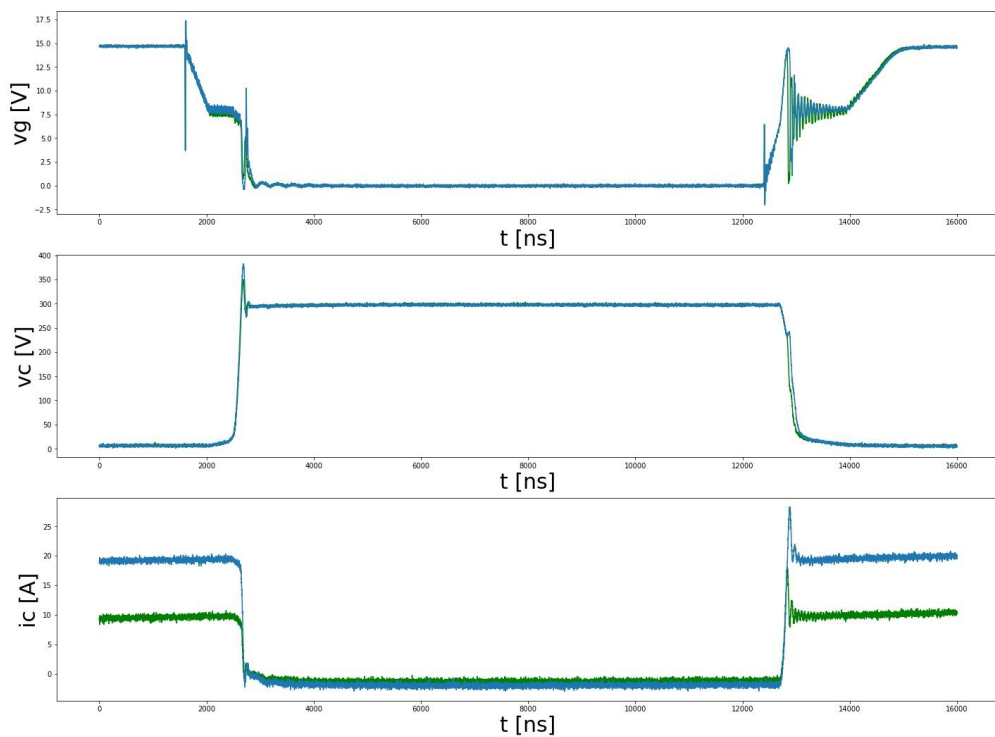


(b) 動作確認回路で取得した波形

図 5-4 シミュレーション波形と実測波形の比較



(a) シミュレーションで取得した波形



(b) 高電圧測定で取得した波形

図 5-5 シミュレーション波形と実測波形の比較

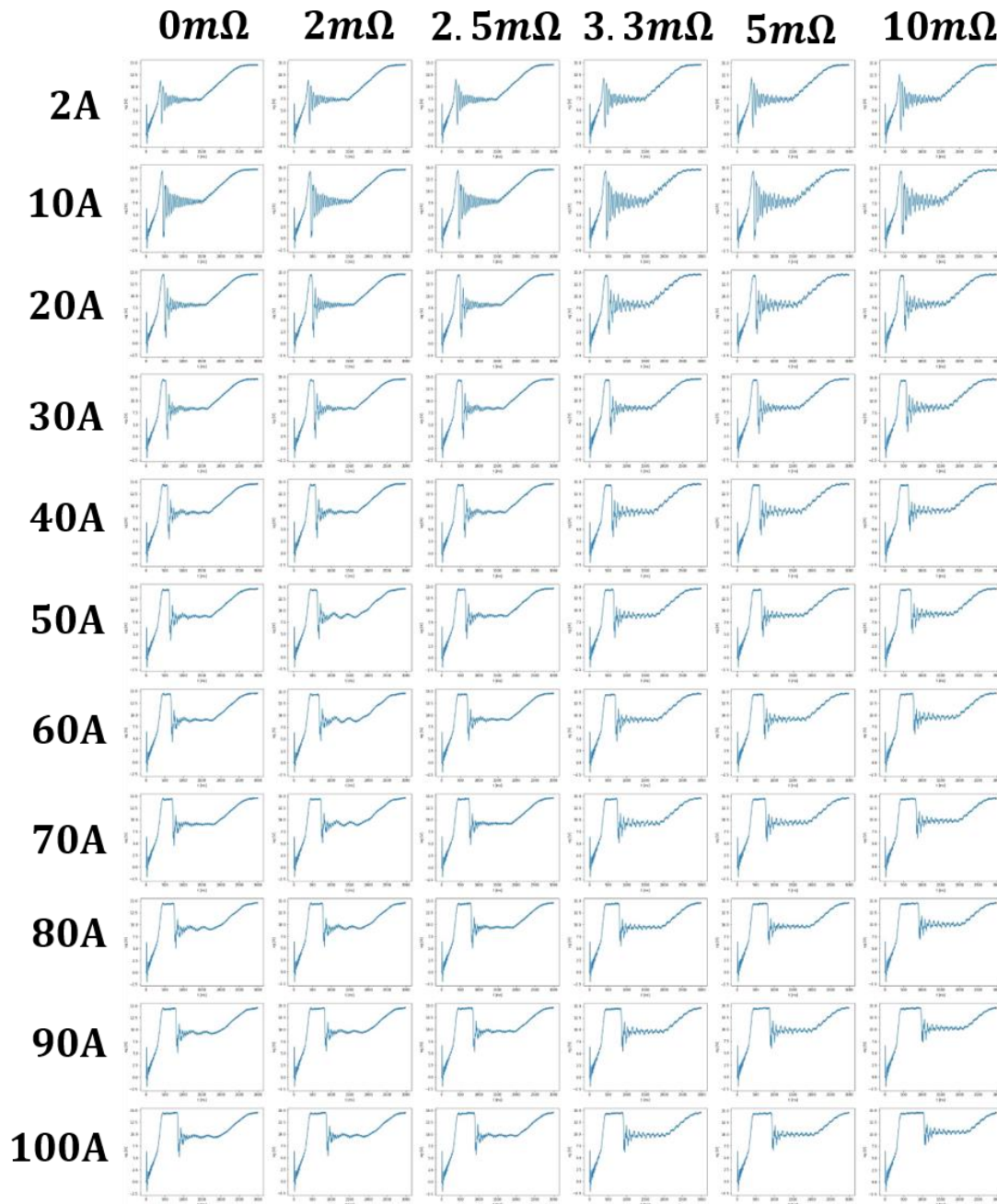


図 5-6 実測ゲート電圧波形の全パターン

ンスが通常より大きくなっている。ターンオンしたときに発生する $\frac{di}{dt}$ はこの寄生インダクタンスと伴ってエミッタ端子の電圧値を一時的上昇させ、同時に C_{GD} に伝わってゲート端子の電圧値が持ち上げられる。しかし、ゲート電圧はゲートドライバによって 15V にクランプしているため、図 5-6 の $I_L \geq 30A$ の波形ではプラトー領域に到達する前にしばらく 15V に頭打ちしていることになっている。図 5-7 から、あらかじめため込ん

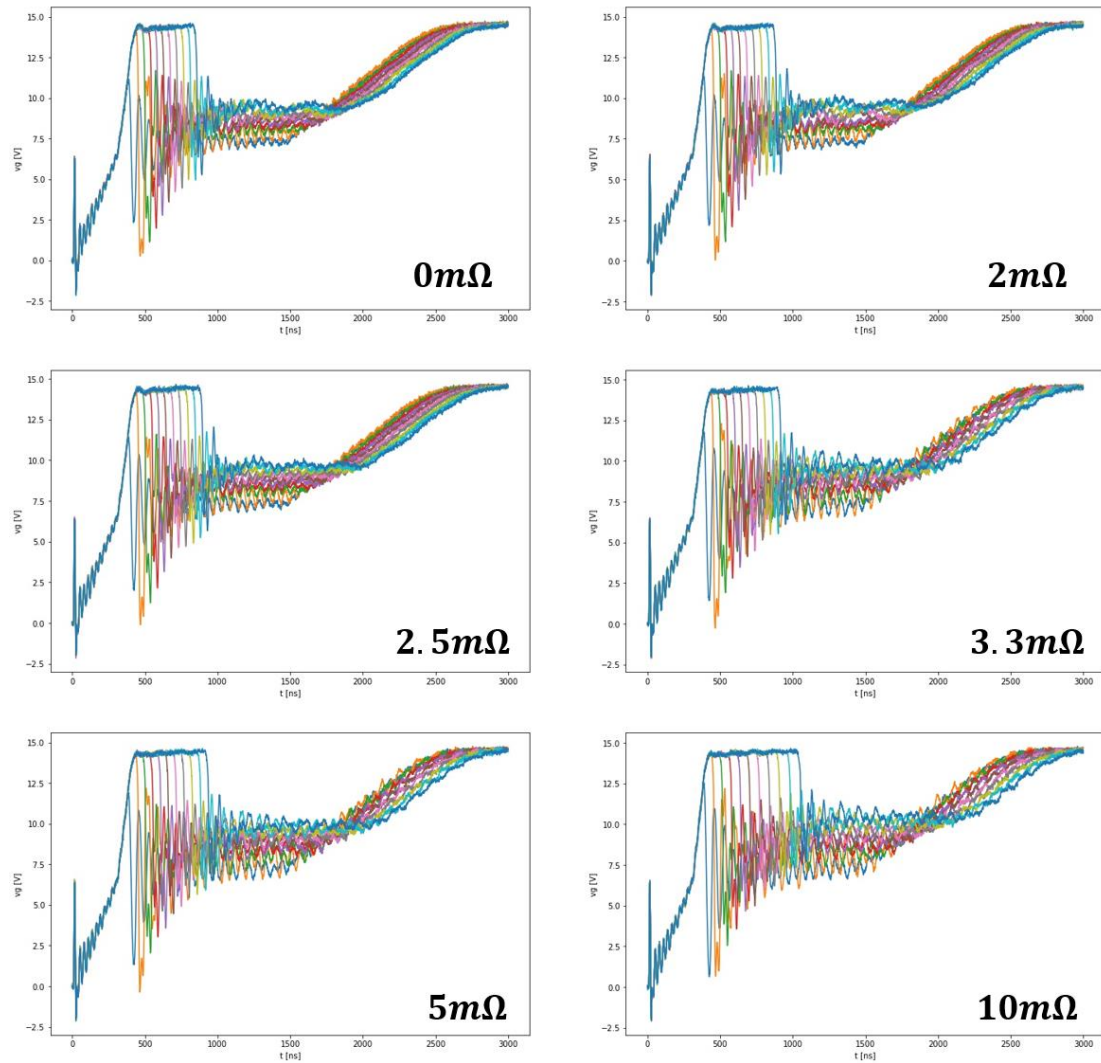


図 5-7 Re が一定のときの異なる L_l によるゲート電圧波形の変化

でいた I_L が大きければ大きいほど、ターンオン後に流れ始めた I_C が I_L の値に到達するまでの時間が遅れることが人間の目でも観測できる。一方、図 5-8 から、 Re の増加はプラトー領域における振動を激しくし、 I_C を流れにくくする効果で似たような遅れがみられるが、低い I_L においてこれらの変化はほとんどみられず、 $I_L \geq 30A$ あたりからようやく人間の目でも観測できるようになっている。したがって、このような Re による微小な波形変化を見分けることができれば、機械学習やニューラルネットワークアルゴリズムを用いる意義となると考えられる。

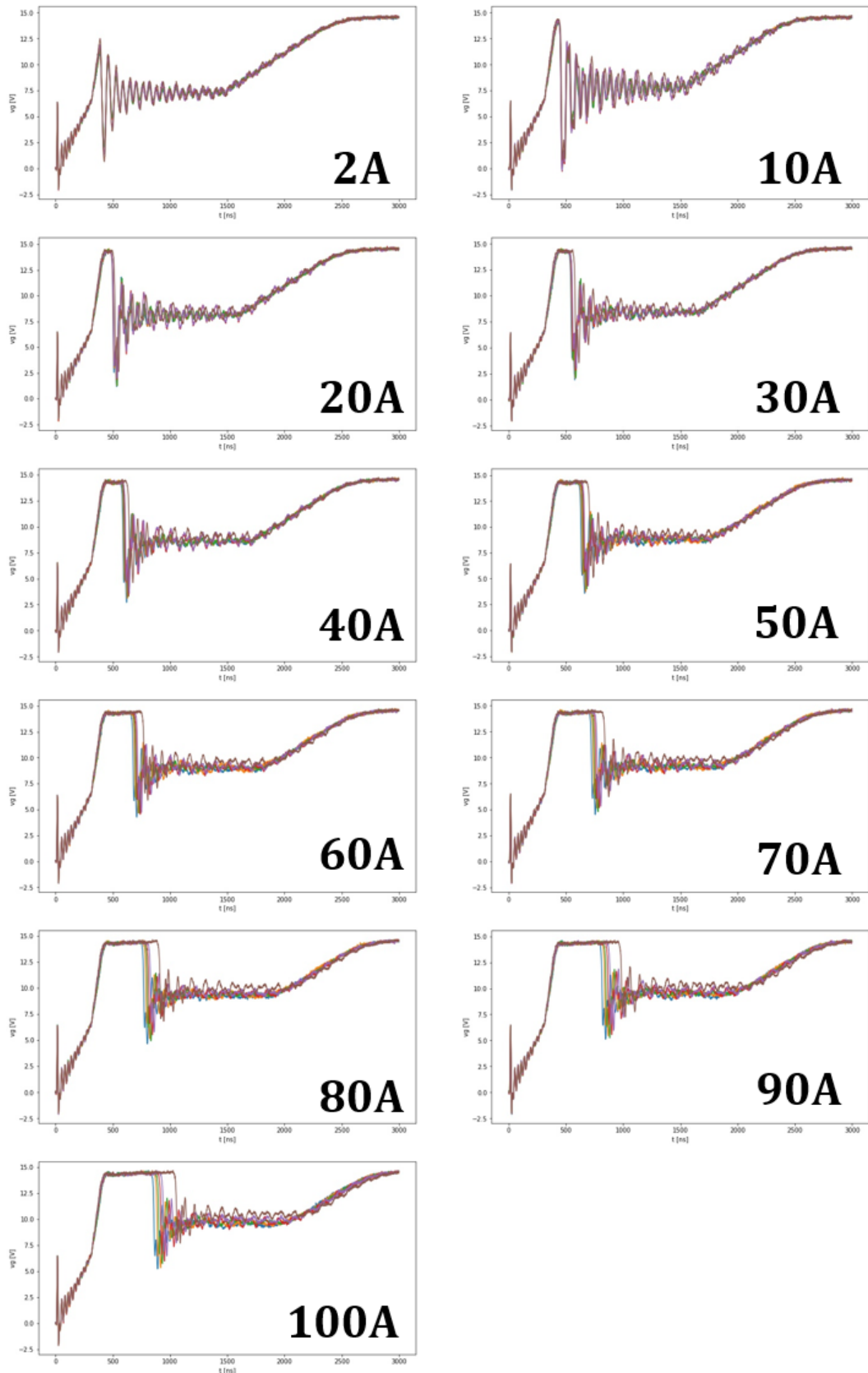


図 5-8 I_L が一定のときの異なる Re によるゲート電圧波形の変化

第6章 実験結果

シミュレーションで取得した波形データのデバイスパラメータの変動幅を五分割し、五クラスの分類問題とする。このとき、クラス番号が大きい順でパワーデバイスの劣化度合が大きいことになっている。波形データの内訳を表 6-1 に示す。これを用いて学習アルゴリズムで学習し、その結果について比較を行う。

表 6-1 シミュレーション波形のラベル付け

0: $R_s < 2\text{m}\Omega$	1: $2\text{m}\Omega \leq R_s < 4\text{m}\Omega$	2: $4\text{m}\Omega \leq R_s < 6\text{m}\Omega$	3: $6\text{m}\Omega \leq R_s < 8\text{m}\Omega$	4: $R_s \geq 8\text{m}\Omega$	合計
6864	5016	5280	6864	6864	30888

0: $V_{th} < 1\text{V}$	1: $1\text{V} \leq V_{th} < 2\text{V}$	2: $2\text{V} \leq V_{th} < 3\text{V}$	3: $3\text{V} \leq V_{th} < 4\text{V}$	4: $V_{th} \geq 4\text{V}$	合計
4752	7128	4752	7128	7128	30888

0: $T_j < 50^\circ\text{C}$	1: $50^\circ\text{C} \leq T_j < 75^\circ\text{C}$	2: $75^\circ\text{C} \leq T_j < 100^\circ\text{C}$	3: $100^\circ\text{C} \leq T_j < 125^\circ\text{C}$	4: $T_j \geq 125^\circ\text{C}$	合計
5616	5616	5616	5616	8424	30888

0: $I_L < 20\text{A}$	1: $20\text{A} \leq I_L < 40\text{A}$	2: $40\text{A} \leq I_L < 60\text{A}$	3: $60\text{A} \leq I_L < 80\text{A}$	4: $I_L \geq 80\text{A}$	合計
7722	5148	5148	5148	7722	30888

一方、学習アルゴリズムの検証に使う実測データについては、同様に表 6-2 に示す。

表 6-2 実測波形のラベル付け

0: $R_e = 0\text{m}\Omega$	1: $R_e = 2\text{m}\Omega$	2: $R_e = 2.5\text{m}\Omega$	3: $R_e = 3.3\text{m}\Omega$	4: $R_e = 5\text{m}\Omega$	5: $R_e = 10\text{m}\Omega$	合計
1100	1100	1100	1100	1100	1100	6600

0: $I_L < 15\text{A}$	1: $15\text{A} \leq I_L < 30\text{A}$	2: $30\text{A} \leq I_L < 45\text{A}$	3: $45\text{A} \leq I_L < 60\text{A}$	4: $60\text{A} \leq I_L < 75\text{A}$	5: $I_L \geq 75\text{A}$	合計
1200	600	1200	600	1200	1800	6600

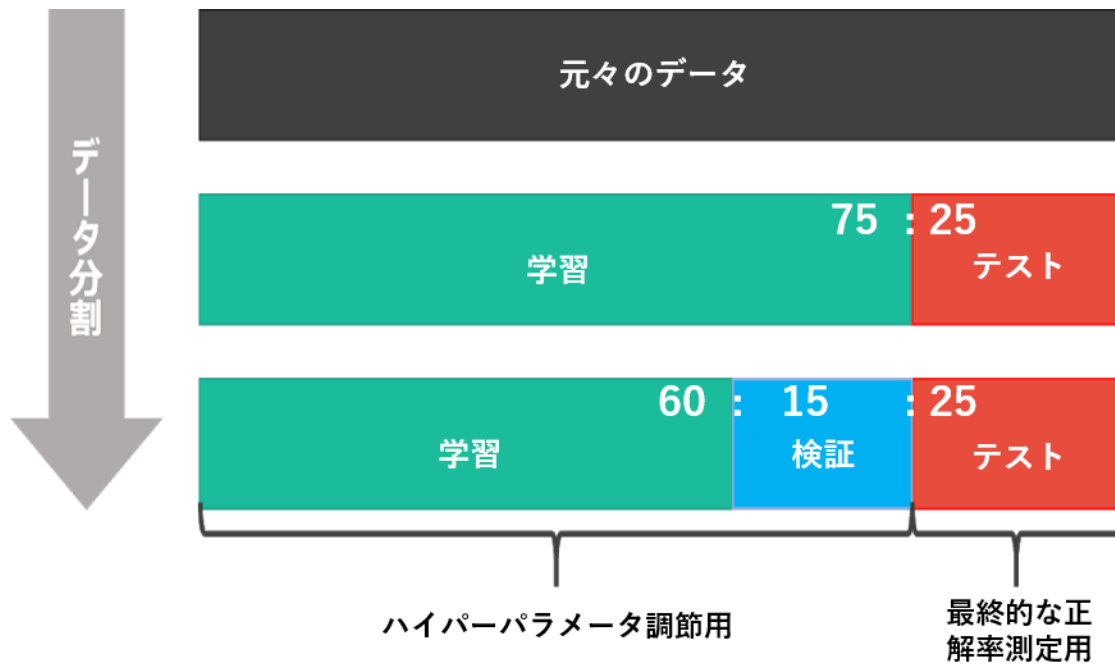


図 6-1 学習データの分割

また、図 6-1 のように 60 : 15 : 25 の比率で学習データと検証データ、テストデータに分割し、正解率や混同行列など学習結果の評価を行う。

6.1 採用した三種類の学習フローの学習結果

6.1.1 PCA+kNN を用いた学習

PCA は元々のデータの特徴量の数までの次元削減が行えるが、まずデータを何次元 ($n_components$) まで削減するかを決める。図 6-2 にシミュレーションデータを第 50 主成分までの累積寄与率を示す。これにより、採用した主成分で元々のデータの何割の情報量を持っているかを表すことができる。例えば、 $n_components=10$ にすると、約 80% の寄与率を得ることができ、 $n_components=15$ すると、約 90% の寄与率を得ることができる。

$n_components=15$ にし、その他はハイパーパラメータをデフォルト (kNN の $k=5$ など) としたときのテストデータの正解率および混同行列を表 6-3 に示す。 V_{th} と I_L に関する分類は 99% の正解率を得ており、 R_s についても約 90% の正解率に達している。また、不正解になった箇所でも、すぐ隣のラベルに分類されている場合がほとんどであることが混同行列からみられる。しかし、 T_j に対する分類の正解率は 50% 以下になっている。

この PCA による学習結果への影響を考察するために、データの可視化および正解率ヒートマップを用いる。第一主成分および第二主成分からなる新たな特徴平面 (寄与率

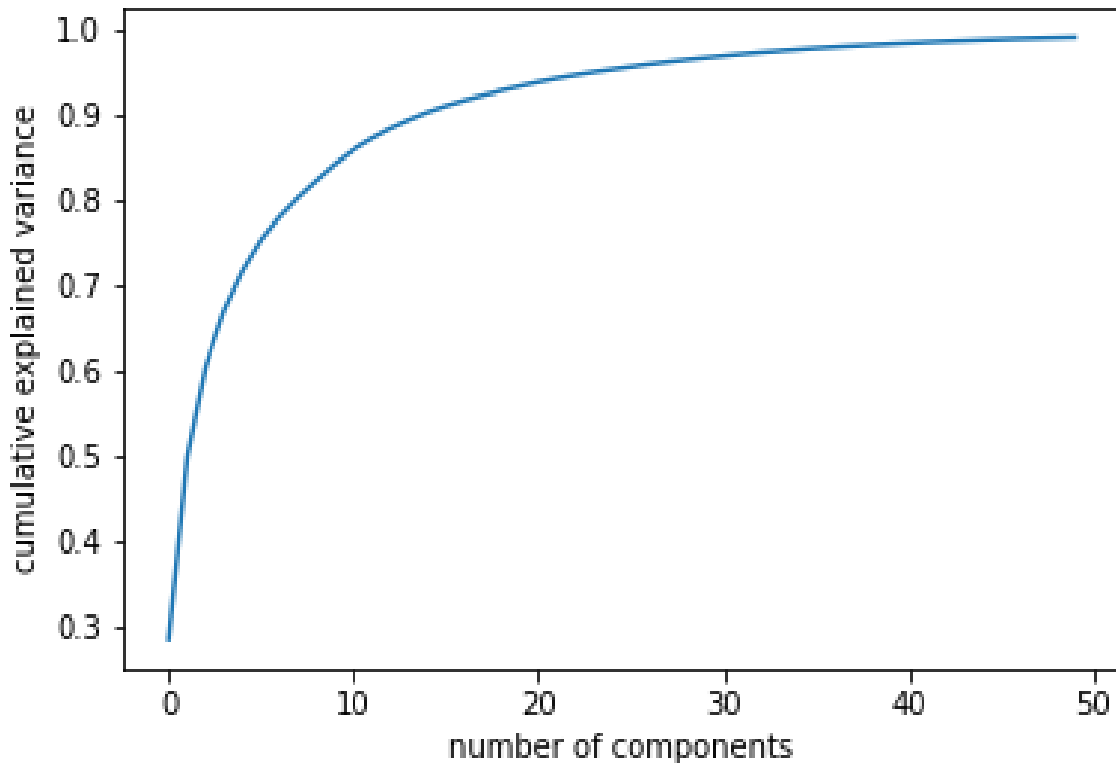


図 6-2 主成分の累積寄与率

約 50%)に射影した結果を図 6-3 に示す。これを図 6-4 のようにラベルごとに色分けを行う。一方、PCA の $n_components=1\sim15$ 、kNN の $k=1,3,5,7,9,11,13,15$ にそれぞれ変えたときの正解率を図 6-5 に示す。

図 6-4 から、PCA による変換は $V_{th} > I_L > R_s > T_j$ の順で効果的な分類がみられる。 V_{th} は図 6-5(b)で $n_components \geq 2$ からほぼ 100%を得ていることがみられる。 I_L については、 $n_components=2$ のとき、ラベル 2~4 のデータが分けられていないため、正解率は約 70%であるが、 $n_components=3$ で約 90%まで上がり、 $n_components \geq 4$ からほぼ 100%を得ていることから、必要な次元数がやや多めに採用することでよい正解率が得られることが 6-5(d)から見て取れる。 R_s では似た傾向がみられるが、6-5(a)から $n_components$ の値に関係なく $k=1$ のときに最も高い正解率を得ていることは、異なるラベルを持つデータは所々で混在し、最も近い一個のデータ以外は別のラベルを持ったデータが存在している場合が多いことを示している。これは T_j についても同様で、結果がより悪くなったことが考えられる。結論として、 V_{th} と I_L は成功したが、 R_s と T_j による変動を特徴として抽出する効果が低いことがみられる。

表 6-3 PCA+kNN での学習結果

Rs 正解率:0.913

予測\実際	0	1	2	3	4	適合率
0	1714	80	0	2	0	0.954
1	73	1118	66	2	0	0.888
2	4	61	1074	168	1	0.821
3	0	6	105	1542	29	0.917
4	0	0	7	67	1603	0.956
再現率	0.957	0.884	0.858	0.866	0.982	

Vth 正解率:0.993

予測\実際	0	1	2	3	4	適合率
0	1170	4	0	0	0	0.997
1	0	1769	24	0	0	0.987
2	0	2	1166	16	0	0.985
3	0	0	0	1782	5	0.997
4	0	0	0	0	1784	1
再現率	1	0.997	0.980	0.991	0.997	

Tj 正解率:0.444

予測\実際	0	1	2	3	4	適合率
0	803	475	43	25	92	0.558
1	450	600	320	58	14	0.416
2	62	409	481	354	76	0.348
3	24	68	414	394	470	0.288
4	24	17	245	652	1152	0.551
再現率	0.589	0.382	0.320	0.266	0.639	

Il 正解率:0.989

予測\実際	0	1	2	3	4	適合率
0	1933	0	0	0	0	1
1	0	1292	0	0	0	1
2	0	0	1252	2	0	0.998
3	0	0	0	1267	66	0.950
4	0	0	0	18	1892	0.990
再現率	1	1	1	0.984	0.966	

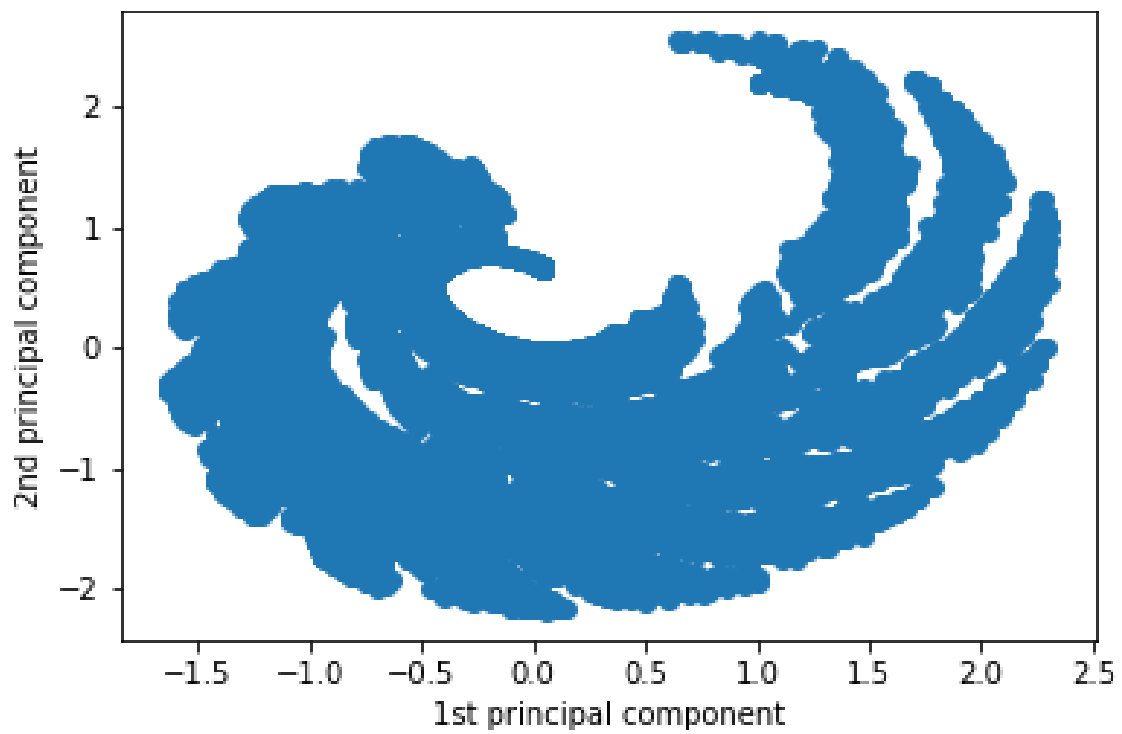


図 6-3 PCA で変換したシミュレーション波形データ

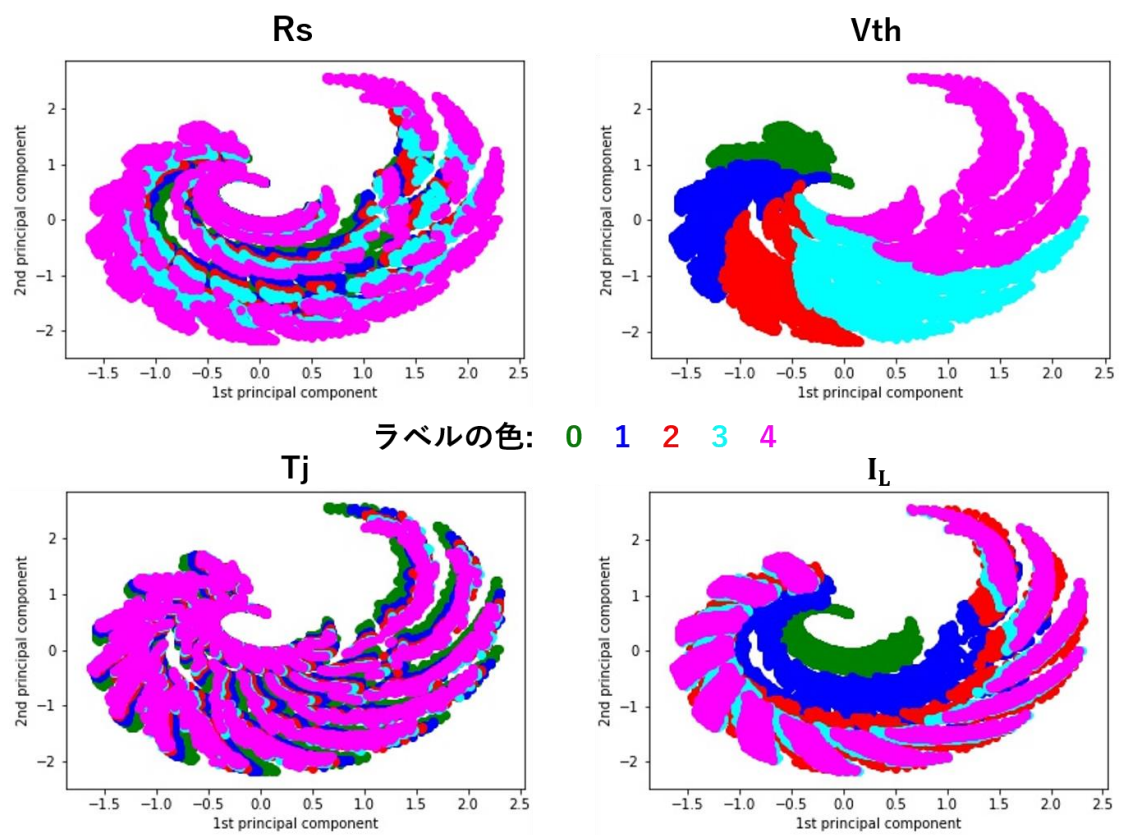
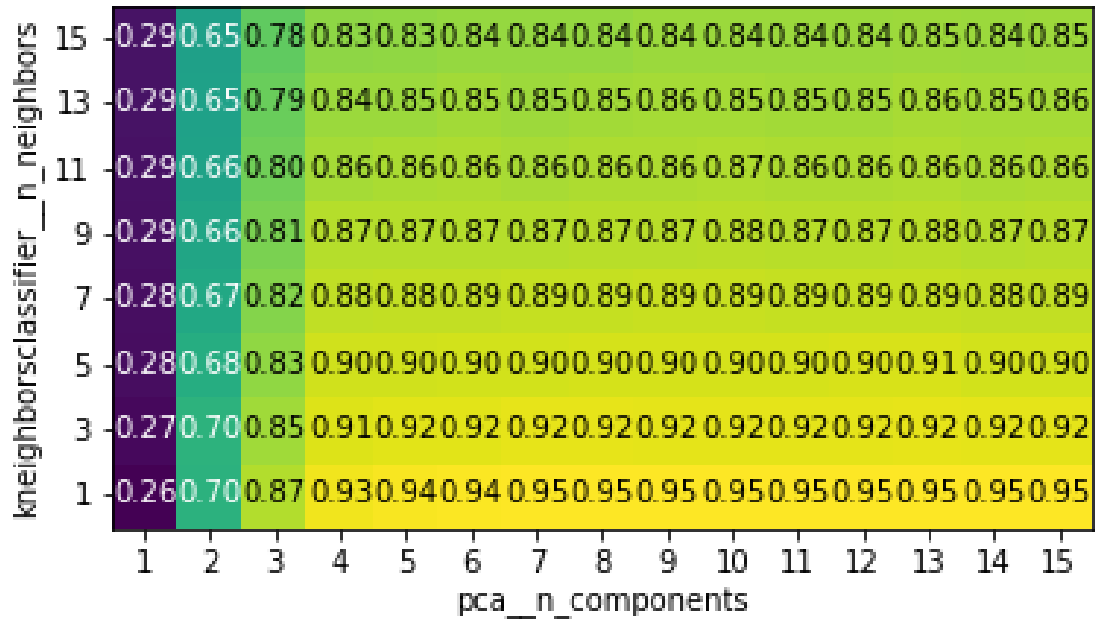
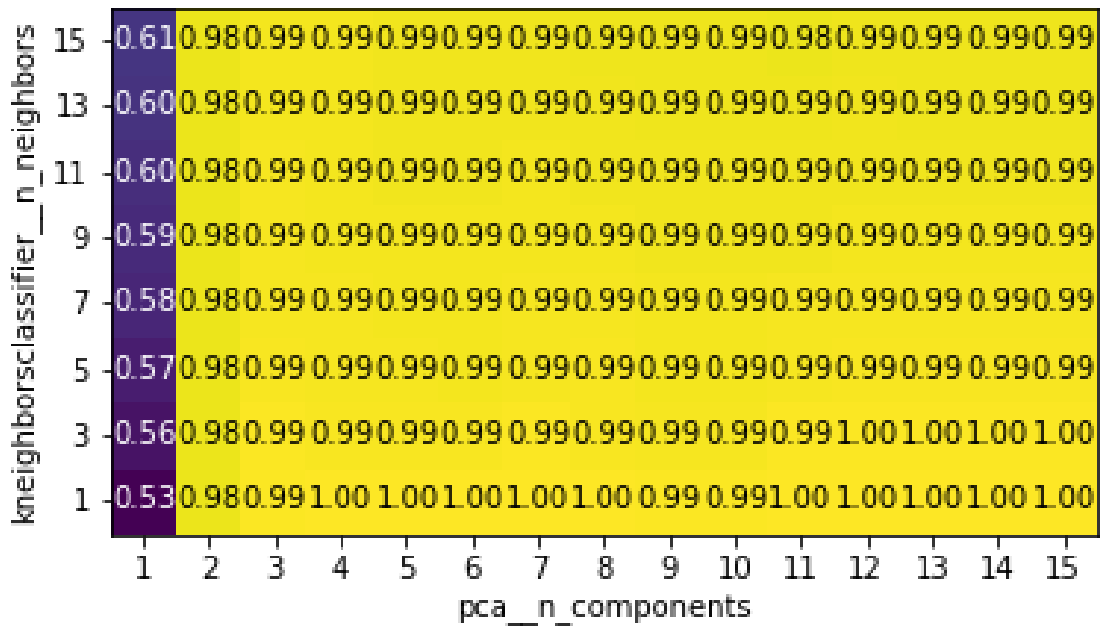


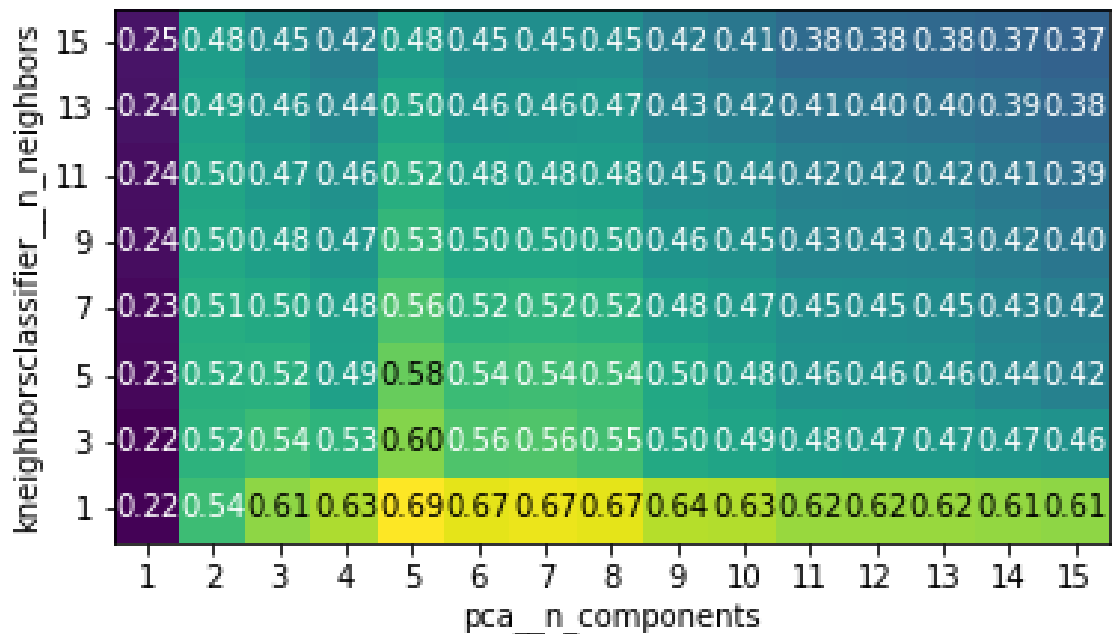
図 6-4 ラベルで色分けした変換後の波形データ



(a) Rsの正解率ヒートマップ



(b) Vthの正解率ヒートマップ



(c) Tjの正解率ヒートマップ

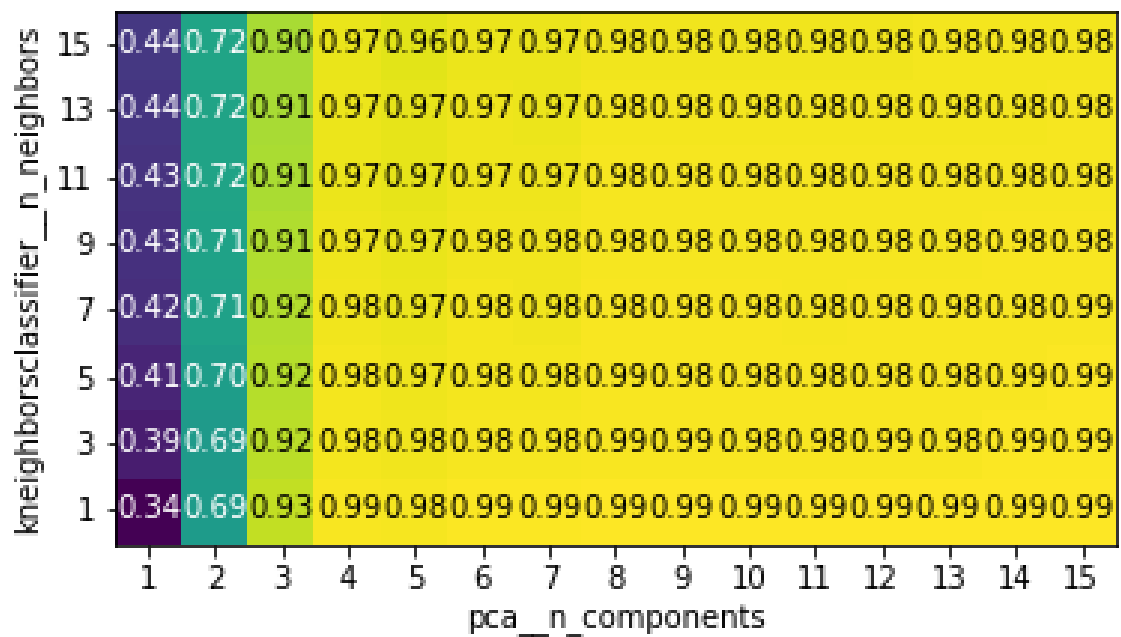
(d) I_Lの正解率ヒートマップ

図 6-5 PCA と kNN ハイパーパラメータごとのテストデータ正解率

6.1.2 LDA+kNN を用いた学習

表 6-4 LDA+kNN での学習結果

Rs 正解率: 0.963

予測\実際	0	1	2	3	4	適合率
0	1720	15	0	0	0	0.991
1	15	1209	38	0	0	0.958
2	0	17	1223	72	0	0.932
3	0	3	58	1628	16	0.954
4	0	4	7	33	1664	0.974
再現率	0.991	0.969	0.922	0.939	0.9	

Vth 正解率: 0.982

予測\実際	0	1	2	3	4	適合率
0	1161	9	0	0	0	0.992
1	6	1732	19	0	0	0.986
2	0	8	1155	43	0	0.958
3	0	0	14	1806	30	0.976
4	0	0	0	10	1729	0.994
再現率	0.995	0.990	0.972	0.971	0.983	

Tj 正解率: 0.441

予測\実際	0	1	2	3	4	適合率
0	697	372	150	108	80	0.495
1	419	487	214	133	115	0.356
2	243	227	525	209	187	0.377
3	181	171	276	436	345	0.309
4	131	133	231	382	1260	0.590
再現率	0.417	0.350	0.376	0.344	0.634	

I_L 正解率: 0.959

予測\実際	0	1	2	3	4	適合率
0	1972	0	0	0	0	1
1	0	1336	6	0	0	0.996
2	0	5	1192	37	0	0.966
3	0	0	30	1101	139	0.867
4	0	0	0	100	1804	0.947
再現率	1	0.996	0.971	0.889	0.928	

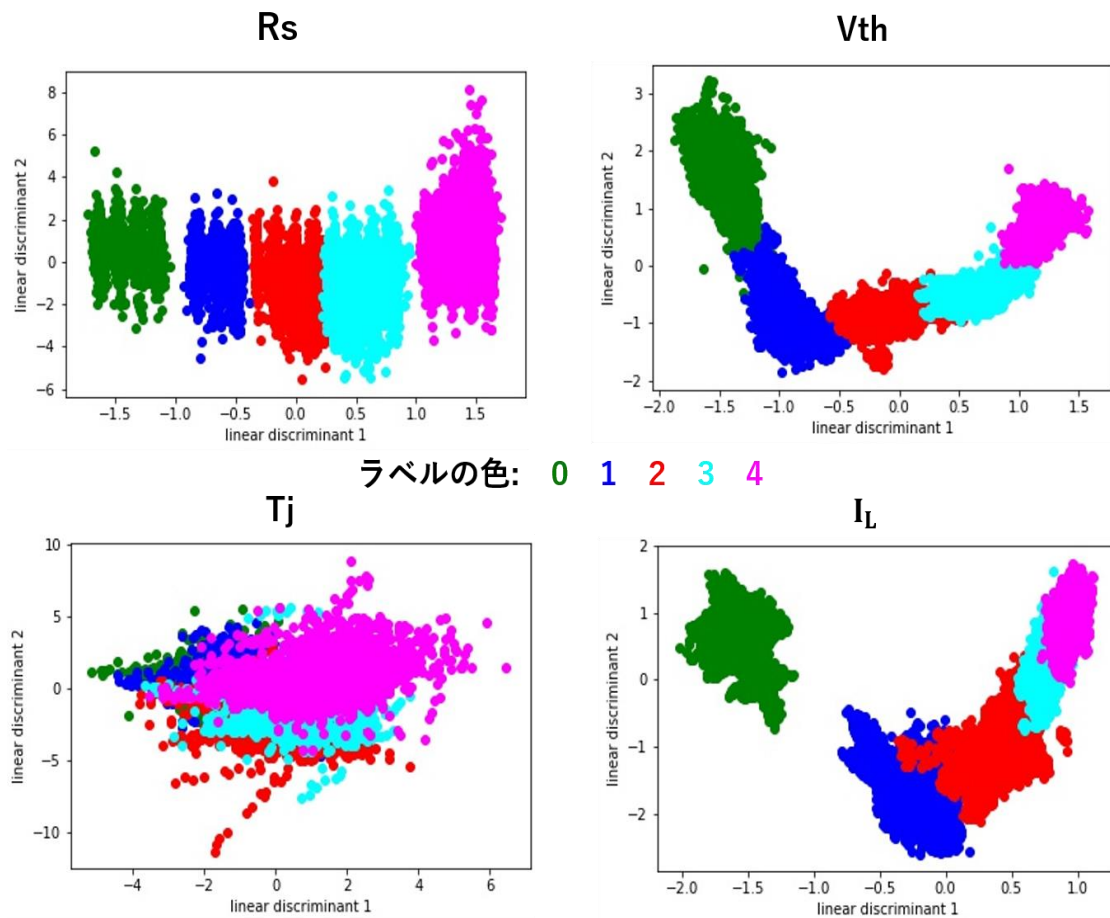
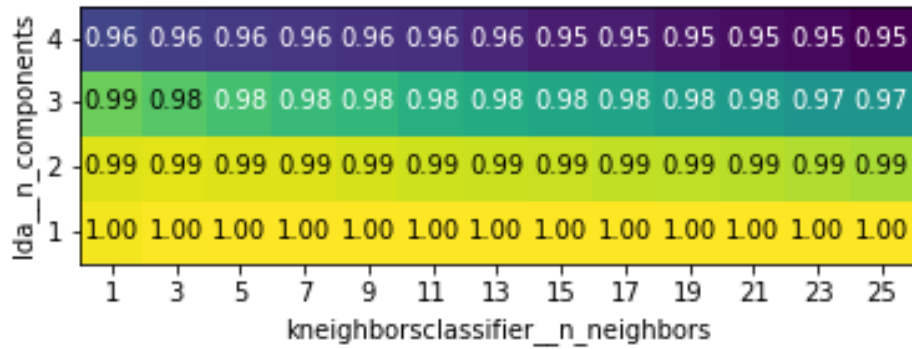
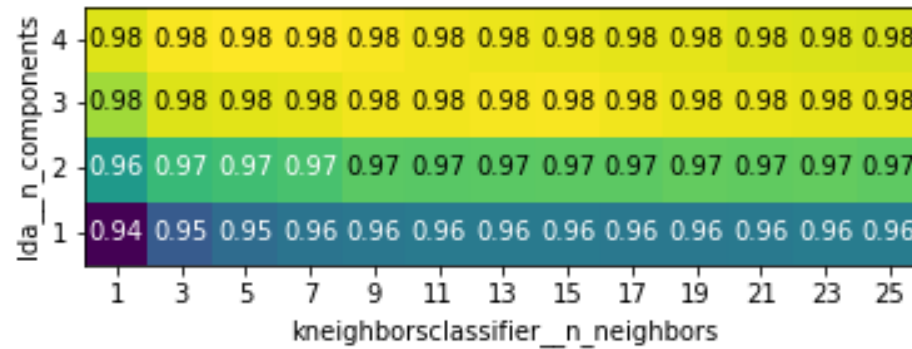


図 6-6 LDA で変換したシミュレーション波形データ

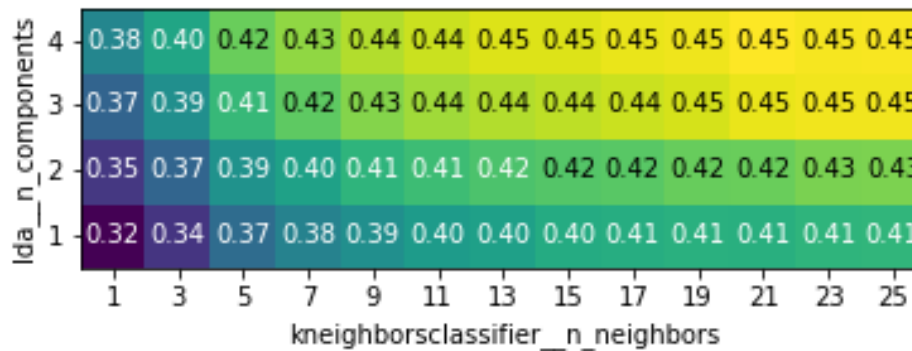
LDA はラベルの数-1 まで次元削減が行える。ラベルの数は 5 のため、変換後の次元数は最大 $n_components=4$ までである。 $n_components=4$ にし、その他はハイパーパラメータをデフォルトとしたときのテストデータの正解率および混同行列、さらににデータの二次元射影図と正解率ヒートマップを同様に表 6-4、図 6-6、図 6-7 に示す。**Rs**と**Vth**、**IL**に関する分類は 95%以上の正解率を得ているが、**Tj**については PCA のときと同様に 50%以下になっている。これは LDA による変換は**Rs**と**Vth**、**IL**についてラベルごとの仕分けに成功し、**Tj**については失敗しているからであることが、図 6-6 から見て取れる。また、図 6-7 から k の値に関係なく、同じ $n_components$ ではほぼ同じ正解率を得ていることも、**Rs**と**Vth**、**IL**について分類が成功していることの裏付けとなっている。



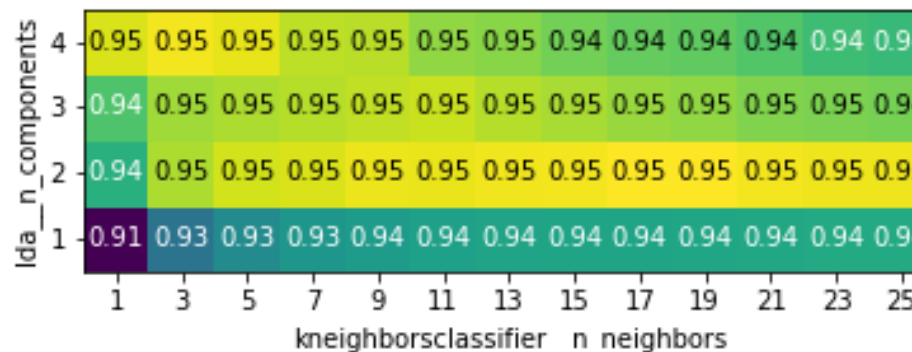
(a) Rsの正解率ヒートマップ



(b) Vthの正解率ヒートマップ



(c) Tjの正解率ヒートマップ



(d) ILの正解率ヒートマップ

図 6-7 LDA と kNN ハイパーパラメータごとのテストデータ正解率

6.1.3 CNN を用いた学習

表 6-5 に用いた CNN モデルを示す。層数、フィルタ数、フィルタサイズ、プーリングサイズ、全結合層の数およびニューロン数などは図 3-5 のモデルを一次元に書き換えた形になっている。このモデルでの学習結果は表 6-6 に示す。CNN を用いた学習は、全てのデバイスパラメータの分類に 90%以上の正解率を得られた。次に、ゲート波形の学習に対して、異なるフィルタサイズによる影響を確認する。フィルタサイズ 5~40 を用いたときの CNN モデルで学習したとき、50 エポックが終了した時点でのテストデータに対する正解率を図 6-8 に示す。結果として、フィルタサイズが 25 のときに学習結果が総合的にもっともよかったことがみられた。

表 6-5 CNN モデル

層 (タイプ)	出力サイズ	重みパラメータ数
conv1d_1 (一次元畳み込み)	(None, 1000, 6)	156
max_pooling1d_1 (一次元最大値プーリング)	(None, 500, 6)	0
conv1d_2 (一次元畳み込み)	(None, 500, 16)	2416
max_pooling1d_2 (一次元最大値プーリング)	(None, 250, 16)	0
flatten_1 (一次元変換)	(None, 4000)	0
dense_1 (全結合)	(None, 120)	480120
dense_2 (全結合)	(None, 84)	10164
dense_3 (全結合)	(None, 5)	425
Total params: 493,281		
Trainable params: 493,281		
Non-trainable params: 0		

表 6-6 CNNでの学習結果

Rs 終了エポック:68 損失:0.102 正解率:0.963

予測\実際	0	1	2	3	4	適合率
0	1663	45	0	0	0	0.974
1	4	1201	54	3	0	0.952
2	0	4	1207	111	0	0.913
3	0	0	33	1629	73	0.939
4	0	0	0	5	1690	0.997
再現率	0.998	0.961	0.933	0.932	0.959	

Vth 終了エポック:86 損失:0.0107 正解率:0.996

予測\実際	0	1	2	3	4	適合率
0	1160	3	0	0	0	0.997
1	0	1764	5	0	0	0.997
2	0	0	1185	13	0	0.989
3	0	0	0	1814	13	0.993
4	0	0	0	0	1765	1
再現率	1	0.998	0.996	0.993	0.993	

Tj 終了エポック:95 損失:0.163 正解率:0.933

予測\実際	0	1	2	3	4	適合率
0	1253	116	7	1	0	0.910
1	36	1318	73	0	0	0.924
2	0	5	1322	87	12	0.927
3	0	0	18	1232	154	0.877
4	0	0	0	12	2076	0.994
再現率	0.972	0.916	0.931	0.925	0.926	

Il 終了エポック:53 損失:0.00740 正解率:0.996

予測\実際	0	1	2	3	4	適合率
0	1882	0	0	0	0	1
1	0	1268	0	0	0	1
2	0	2	1260	0	0	0.998
3	0	0	0	1317	27	0.980
4	0	0	0	0	1966	1
再現率	1	0.998	1	1	0.986	

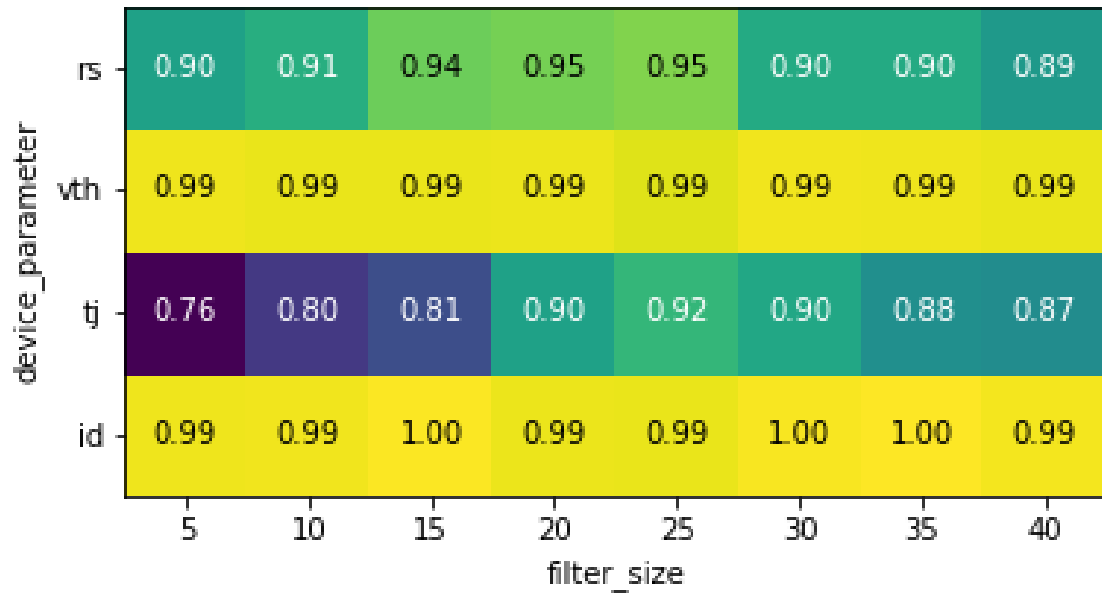


図 6-8 CNN フィルタサイズによる学習結果への影響

6.2 A/D 変換器性能による学習結果への影響

CNN を用いた学習がもっともよかったことが確認したため、続いて実装するにあたって必要な A/D 変換器性能を調べる。今までのシミュレーションデータは 1ns 刻みで、小数点 15 桁の電圧値のままで学習を行ったが、A/D 変換器のサンプリング間隔および分解能が [1,2,4,5,10,20,25,40,50]ns と [12,10,8,6]bit のときのデータで CNN モデルに学習させる。それぞれの条件における学習結果を図 6-9 の正解率ヒートマップおよび図 6-10 のフィッティング曲線に示す。**Vth**と**I_L**については正解率がほとんど下がらず、常に 90%以上を維持している。一方、**Rs**と**Tj**の正解率の低下は A/D 変換器性能による影響は明らかである。今回用いたシミュレーションデータでは 8bit 以上の分解能であれば、10ns より短いサンプリングの場合、**Rs**が約 90%、**Tj**が約 70%の正解率を得ることができる。

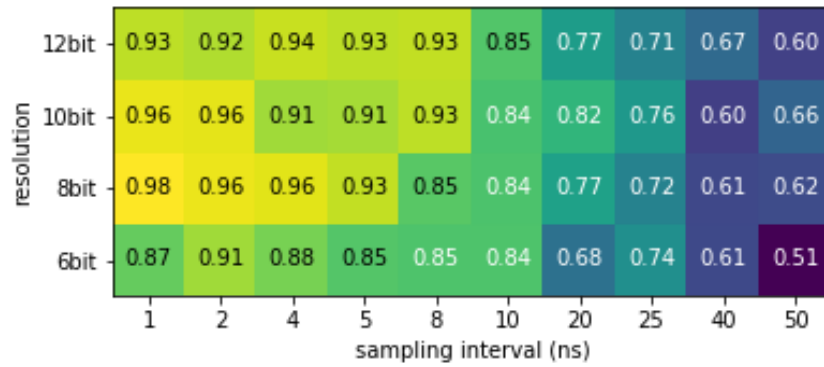
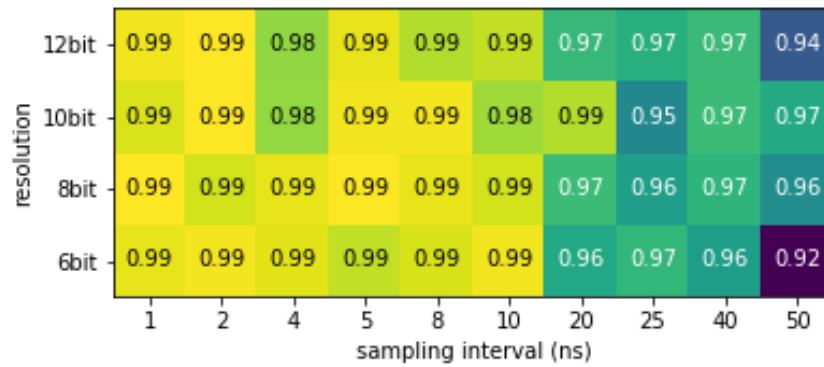
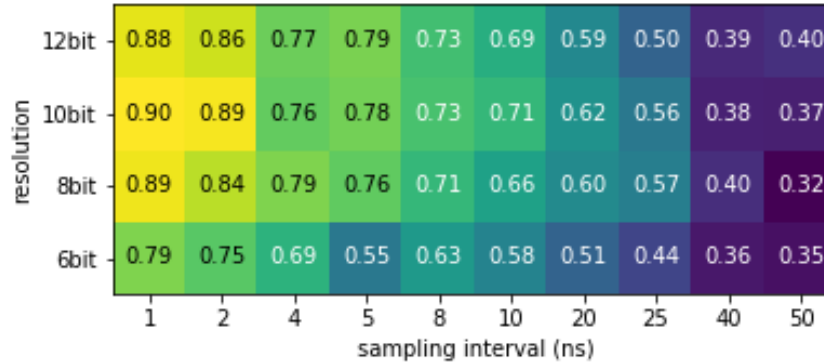
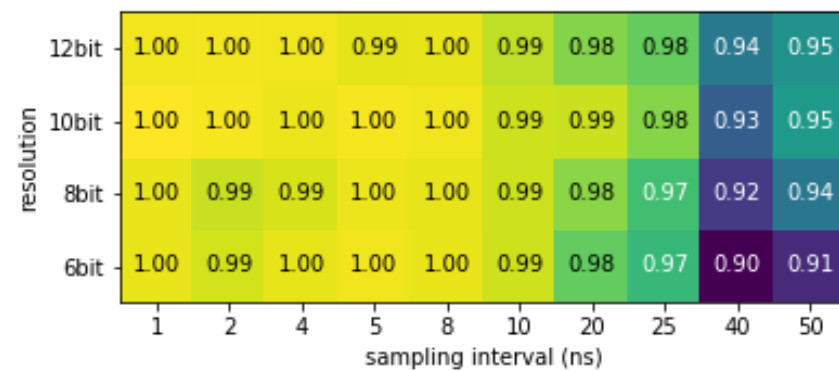
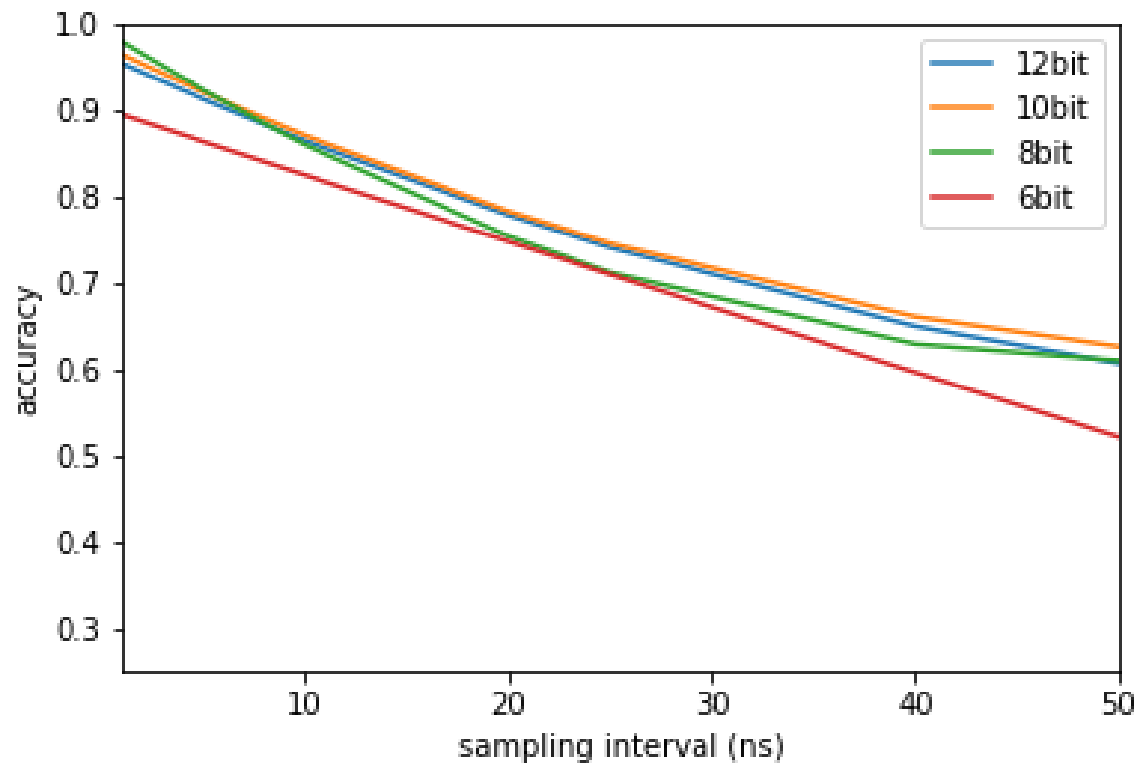
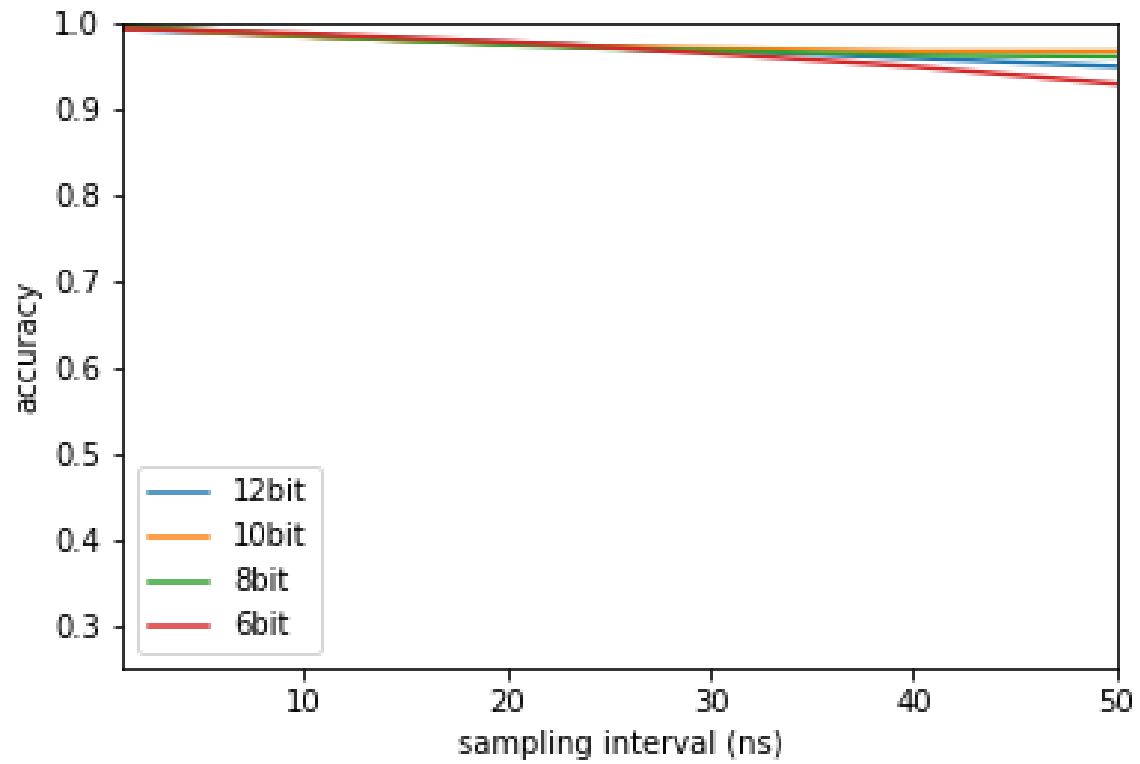
(a) R_s の正解率ヒートマップ(b) V_{th} の正解率ヒートマップ(c) T_j の正解率ヒートマップ(d) I_L の正解率ヒートマップ

図 6-9 A/D 変換器性能による正解率依存

(a) R_s の正解率依存(b) V_{th} の正解率依存

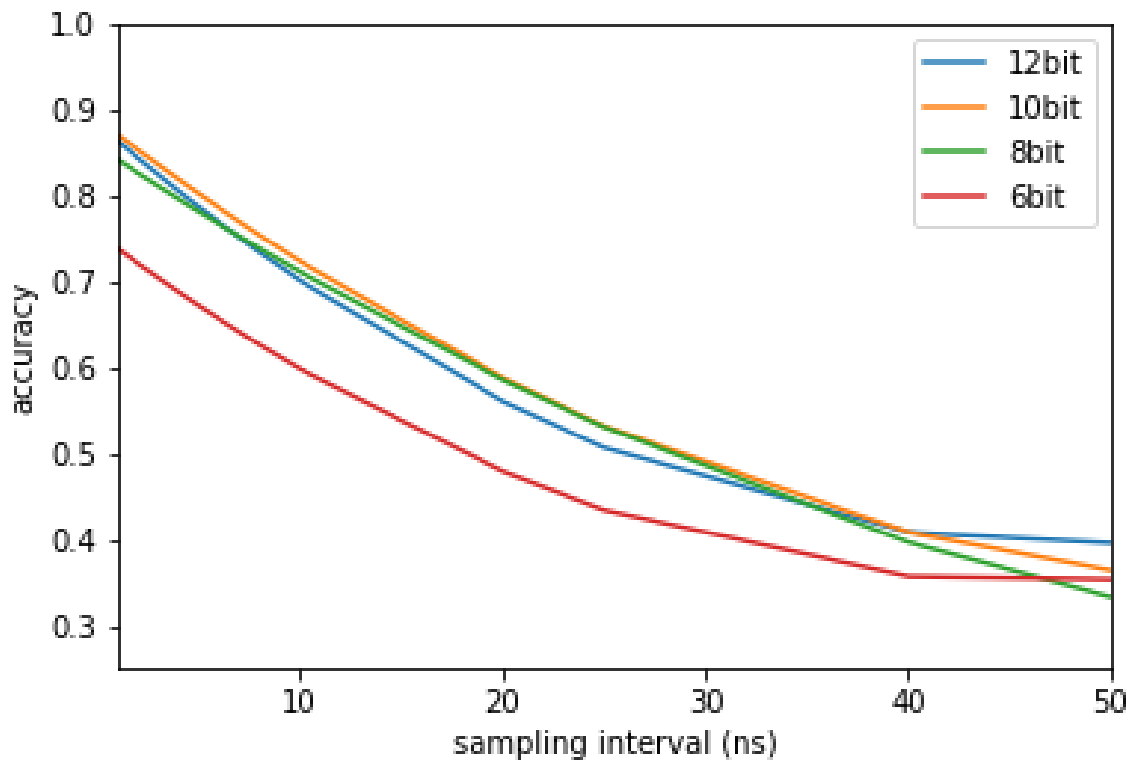
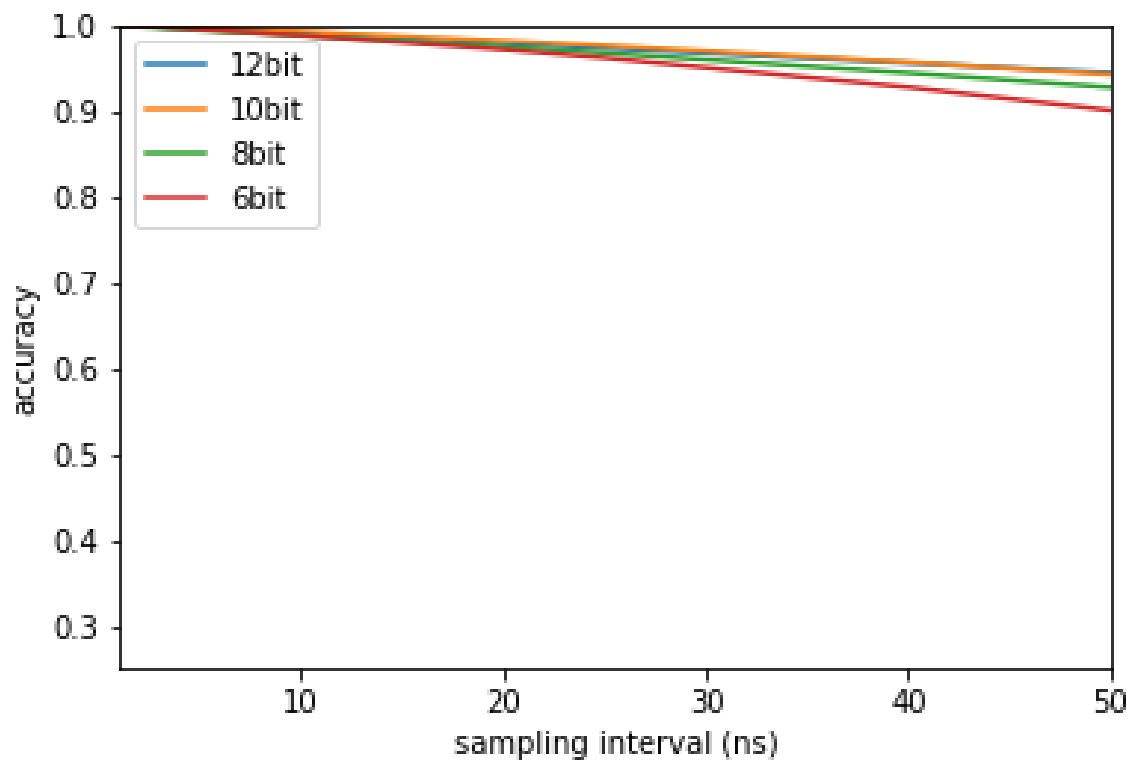
(c) T_j の正解率依存(d) I_L の正解率依存

図 6-10 正解率のサンプリング間隔と分解能依存

6.3 高電圧測定データを用いた検証

今までの実験で得られた結果を高電圧測定で取得した波形データに適用する。つまり、表 6-5 の CNN モデル、8bit の分解能、10ns のサンプリング間隔で学習を行う。このときのテストデータの正解率および混同行列を表 6-7 に示す。**Re**と**I_L**の学習はどちらもほぼ100%の正解率を得ている。これはシミュレーションデータと比べて、変動するデバイスパラメータが少ないため、比較的に分類しやすいことが原因であると考えられる。

表 6-7 高電圧測定データの学習結果

Re 終了エポック:74 損失:0.0225 正解率:0.995

予測\実際	0	1	2	3	4	5	適合率
0	275	3	0	0	0	0	0.989
1	0	265	0	0	0	0	1
2	0	0	264	0	0	0	1
3	0	0	0	262	4	0	0.985
4	0	0	0	1	279	0	0.996
5	0	0	0	0	0	297	1
再現率	1	0.989	1	0.996	0.986	1	

I_L 終了エポック:213 損失:0.000223 正解率:1

予測\実際	0	1	2	3	4	5	適合率
0	315	0	0	0	0	0	1
1	0	144	0	0	0	0	1
2	0	0	272	0	0	0	1
3	0	0	0	155	0	0	1
4	0	0	0	0	298	0	1
5	0	0	0	0	0	466	1
再現率	1	1	1	1	1	1	

この学習結果は妥当かどうかを検証するために、一部ラベルを持つデータを取り除いたときの分類テストを行う。例えば、**Re=2.5mΩ**のときのデータが含まれずに、残りのデータで学習したモデルは、取り除いたデータを残りのどの**Re**ラベルに分類するかテストした。表 6-8 のテスト結果から、1100 個の**Re=2.5mΩ**ときのデータのうち、701 個が**Re=2mΩ**に、398 個が**Re=3.3mΩ**に、1 個だけ**Re=5mΩ**に分類されるという結果になった。その他の**Re**ラベルを持つデータで同じテストをしても、すぐ隣の**Re**ラベルに分類され、離れた**Re**ラベルに分類されることはほとんどない。これにより、CNN による

学習はデバイスパラメータの変動による波形への影響を特徴として学習できており、学習データに含まれなかったデバイスパラメータの値についても推定可能であることを示せていると考えられる。

表 6-8 学習に含まれないデータの分類テスト

含まれないデータが 分類されるクラス	含まれないデータ(n=1100)						
		0mΩ	2mΩ	2.5mΩ	3.3mΩ	5mΩ	10mΩ
	0mΩ		434	0	0	0	0
	2mΩ	1100		701	0	0	0
	2.5mΩ	0	666		190	0	0
	3.3mΩ	0	0	398		815	53
	5mΩ	0	0	1	760		1047
	10mΩ	0	0	0	150	285	

6.4 実験結果のまとめと考察

表 6-9 実験結果のまとめ

	Rs/Re	Vth	Tj	I _L
PCA+kNN	△	◎	×	◎
LDA+kNN	◎	◎	×	○
CNN	◎	◎	○	◎

◎:>95% ○:≥90% △:<90% ×:<70%

採用した三種類の学習フローの学習結果は表 6-9 にまとめる。本研究はパワーデバイスがターンオン時のゲート端子から電圧波形を取得し、それを時系列データとして学習アルゴリズムに適用することを提案した。以上の結果により、さまざまなデバイスパラメータが同時に変動する中でも、推定対象となったデバイスパラメータによる変動だけ抽出し、結果としてパワーデバイスの劣化を検出することが可能であることを示した。また、CNN を用いた学習に必要な計算をクラウドサーバーなどのメインシステムに任せ、出来上がった CNN モデルをゲートドライバにインプットすることで、ローカルでの計算コストは分類を行う際のみ必要となると考えられる。しかし、同時に以下の課題がみられた。

デバイスパラメータによる波形への感度の違い

本研究における分類問題は、表 6-1 および表 6-2 のようにデバイスパラメータの変動幅をそのまま分割して複数のクラスに設定した。抵抗値や温度に対しての分類は、どこまでの条件下では高い正解率が得られ、どこから正解率が悪くなるかの境界を明らかにすることができた。しかし、電流値やしきい値による変動は非常に簡単に分類できており、データの取得条件を悪くしても調べる範囲内では正解率がほとんど下がらなかった。値の変動幅を小さくするか、クラスの数を増やすことなど、分類問題をより厳しくすることが可能であることが考えられる。

ハイパーパラメータのチューニング

学習に用いたアルゴリズムは、それらのハイパーパラメータをほとんどデフォルトにしたまま行った。デフォルト値はライブラリの開発者らがどんな問題でも一般的にうまくいくように設定したものであり、本研究の分類問題についての正解率を全体的に底上げするために、アルゴリズムを特化したパイパーパラメータにチューニングする必要がある。特に畳み込みニューラルネットワークについては、畳み込み層の層数やフィルタ数、プーリング層のサイズ、全結合層の数およびニューロン数など、学習結果にもっとも直結しやすいパイパーパラメータによる効果は今後調べる必要がある。

劣化したパワーデバイスの波形データの取得

本研究で行われた高電圧測定は、IGBT のエミッタ端子に外付け抵抗を付けることで劣化を模擬した。本来パワーデバイスの劣化を調べるためには、パワーサイクル実験を行い、複数のパワーデバイスを実際に劣化させる必要があり、これには多くの時間と部品を要する。しかし、より現実的なデータを得られれば、本研究での提案手法の有効性が増すだろうと思われる。

第7章 結論

パワーデバイスの経年劣化によるパワーエレクトロニクス機器への悪影響は深刻な問題となっている。従来はパワーデバイスの強電部からの直接センシング手法が提案されているが、コストが高いなどの問題があった。

本論文では、パワーデバイスの劣化状況を弱電部から推定するため、機械学習やニューラルネットなどを用いた手法を提案し、シミュレーションや実測で取得した波形データに適用することでその手法の有効性を示した。

第2章でダブルパルススイッチングテストを紹介し、ダブルパルス回路の基本構成およびその動作について説明した。また、パワーデバイスがターンオン時のスイッチング特性について詳細を示した。

第3章では機械学習やニューラルネットワークのオープンソースライブラリを取り上げた。主成分分析、線形判別分析、畳み込みニューラルネットワークなどの特徴抽出アルゴリズムや、 k -最近傍法や人工ニューラルネットワークなどの分類アルゴリズムの仕組みを簡略的に説明した。

第4章ではパワーデバイスの劣化を推定するための先行研究について説明し、その問題点を明らかにした。それらの課題に踏まえて、本研究で提案した手法を述べた。

第5章ではシミュレーションデータと実測データの取得方法を示し、異なるデバイスパラメータのときに取得した波形データを比較した。

第6章では取得した波形データがさまざまな条件下で提案手法に適用したときの学習結果を示した。

参考文献

- [1] N. Valentine, D. Das and M. Pecht, “Failure Mechanism of Insulated Gate Bipolar Transistors (IGBTs),” 2015 NREL Photovoltaic Reliability Workshop, Feb. 24-27 2015.
- [2] N. Vichare and M. Pecht, "Prognostics and Health Management of Electronics," IEEE Transactions on Components and Packaging Technologies, **29**, 2006, 222-229
- [3] M. Pecht, “Prognostics and Health Management of Electronics,” Wiley-Interscience, 2008.
- [4] N. Patil, S. Menon, D. Das and M. Pecht, "Anomaly Detection of Non Punch Through Insulated Gate Bipolar Transistors (IGBT) by Robust Covariance Estimation Techniques," Reliability Safety and Hazard (ICRESH) 2010 2nd International Conference on, pp. 68-72, 2010.
- [5] N. Baker, S. M. Nielsen, F. Iannuzzo and M. Liserre, “Online Junction Temperature Measurement Using Peak Gate Current,” Applied Power Electronics Conference and Exposition (APEC) 2015 IEEE, pp. 1270-1275, 2015.
- [6] Kikakurui.com, “JIS Z 8115:2000 ディペンダビリティ（信頼性）用語.”
<http://kikakurui.com/z8/Z8115-2000-01.html>
- [7] J. Chen, and W. T. Ng. “Design trends in smart gate driver ICs for power MOSFETs and IGBTs.” IEEE International Conference on Asic, 2017, pp.112-115.
- [8] 山本真義（2015）「インバータの置き換え簡易テスト回路「ダブルパルス回路」」, 『トランジスタ技術』 2015年3号, p.152-154, CQ出版社.
- [9] Vrej Barkhordarian, “アプリケーション・ノート：AN-1084”
<https://www.infineon.com/dgdl/AN-1084.pdf?fileId=5546d46256fb43b301574c5f72917bfc>, International Rectifier Corporation.
- [10] Andreas C. Muller, Sarah Guido（2017）『Pythonではじめる機械学習』（中田秀基 訳）オライリー・ジャパン.
- [11] Aurelien Geron（2018）『scikit-learnとTensorFlowによる実践機械学習』（長尾高弘 訳）オライリー・ジャパン.
- [12] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P.: ‘Gradient-based learning applied to document recognition’, Proc. IEEE, 1998, 86, (11), pp. 2278–2324
- [13] Francois Chollet（2018）『PythonとKerasによるディープラーニング』（巢籠悠輔 訳）マイナビ出版.
- [14] “scikit-learn: machine learning in Python — scikit-learn 0.20.2 documentation”
<https://scikit-learn.org/stable/index.html>

- [15] “Home - Keras Documentation” <https://keras.io/ja/>
- [16] Y. Xiong, Xu. Cheng, Z. Shen, C. Mi, H. Wu, and V. Garg, “Prognostic and Warning System for Power-Electronic Modules in Electric, Hybrid Electric, and Fuel-Cell Vehicles,” IEEE Trans. on Industrial Electronics, vol. 55, no. 6, pp. 2268-2276, June 2008.
- [17] Edwin Sutrisno, “Fault detection and prognostics of insulated gate bipolar transistor (IGBT) using a k-nearest neighbor classification algorithm,” MS dissertation, Department of Mechanical Engineering, College Park, MD, 2013.

本研究に関する発表

- [1] 羅 揚, 宮崎 耕太郎, イスラム マーフズル, 高宮 真, 桜井 貴康, "ゲート電圧波形の解析によるパワーデバイスの異常検知の検討," 電子情報通信学会総合大会, A-3-2, 東京, 2018 年 3 月.

謝辞

本研究を進めるにあたって協力をいただいた様々な方にこの場を借りてお礼を申し上げます。

指導教官の桜井貴康先生には研究内容に関する手厚いご指導を頂いたことは言うに及ばず、エレクトロニクスを通して学术界・産業界を俯瞰する先生の視野の一端に触れ、共有できたことは研究生活における非常に大きな収穫となりました。心より御礼申し上げます。

高宮真先生には折に触れて研究上の様々なアドバイスを頂き、大変有益な議論をしていただきました。また研究に対する向き合い方を始め、学会発表や論文執筆等のアドバイスを頂きました。感謝申し上げます。

宮崎耕太郎氏には研究に関するお力添えを頂き、数多くの議論を通じてアドバイスを頂きました。大変ありがとうございました。

研究員の崔通氏、森時彦氏、邱浩氏には日ごろから研究室生活のサポートをしていただいたばかりでなく、回路技術全般の様々な知見を教授していただきました。また元助教 Islam A.K.M. Mahfuzul には初期研修で大変多くの指導を頂きました。奈良裕子氏、小島里佳氏、平井祐未子氏、天野文子氏、黒田舞氏には研究室での活動を続けていく上での数々のサポートを頂きました。皆様に心よりお礼申し上げます。

元博士課程の染谷晃基氏、呉仲祥氏、李承俊氏、および現博士課程の山内善高氏には折に触れて相談に乗っていただいたばかりでなく、普段の交友の中で楽しい研究室生活をサポートしていただきました。同期の加賀谷司氏には何度も締め切り直前の追い込みを共にしていただき、諦めないことの大切さを教わりました。また後輩の茨城亮太郎氏、福留環氏、呉承軒氏、王叡智氏にも大変お世話になりました。皆様に心よりお礼申し上げます。

最後に、大学院進学と東京での生活を支えてくれた両親と姉には感謝してやみません。ありがとうございました。